

Aplikasi Pembelajaran Tenses dan Passive Voice berbasis Android

Makmun

Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Gunadarma
JL. Margonda Raya No. 100 Depok Jawa Barat
Makmun@staff.gunadarma.ac.id

Abstrak

Kemajuan teknologi komunikasi terutama smartphone tumbuh dengan pesatnya di seluruh penghujung dunia. Lembaga riset Emarketer bahkan memperkirakan bahwa pada 2018 akan ada 100 juta orang pengguna smartphone. Saat ini, smartphone berbasis android menjadi solusi untuk mengakses ilmu dan informasi. Mempelajari Tenses dan Passive Voice adalah hal yang paling mendasar bagi seorang pemula yang ingin memahami Bahasa Inggris. Mempelajari Tenses dan Passive Voice dengan membuka buku-buku bahasa Inggris dirasakan kurang praktis dan kurang interaktif. Dengan menggunakan platform Android, aplikasi ini diharapkan dapat membantu masyarakat khususnya anak-anak dan remaja sebagai media alternatif dalam sistem pembelajaran tanpa batasan waktu dan tempat. Perangkat lunak yang digunakan dalam pembuatan aplikasi ini adalah Android-SDK (Software Development Kit), paket instalasi JDK (Java Development Kit), Eclipse Indigo, dan aplikasi ini diuji coba pada emulator Android.

Kata Kunci : Aplikasi, Bahasa Inggris, Tenses, Passive Voice, Android

Pendahuluan

Latar Belakang

Secara psikologis, anak-anak senang dengan sesuatu yang interaktif dan nalarnya akan berjalan lebih cepat jika dia berinteraksi secara langsung dengan media yang dia senangi. Untuk itulah smartphone berbasis android menjadi alat yang dapat membantu siapapun yang ingin mempelajari tenses dengan sistematis [1].

Metode yang digunakan penulis dalam penulisan ilmiah ini adalah studi pustaka (pemahaman literatur) dan internet. Dimulai dari mengumpulkan data-data mengenai Android, tutorial-tutorial yang berkaitan dengan pembuatan aplikasi android serta pembahasan mengenai tenses dan passive voice.

Spesifikasi perangkat keras yang digunakan penulis untuk membuat aplikasi ini adalah sebagai berikut : Processor Intel(R) Core(TM) Duo CPU T5670 @ 1.80 Ghz, VGA Mobile Intel(R) 4 Series Express Chipset Family, RAM 2 GB SDRAM, Monitor 14", Harddisk 160 GB Spesifikasi perangkat lunak yang digunakan penulis untuk membuat aplikasi ini adalah se-

bagai berikut : Sistem Operasi Windows7, Eclipse Java Indigo 3.7.0 win32 dengan ADT 21.0.0, Java JDK 1.7.0, Android SDK dan ADV windows.

Tinjauan Pustaka

Tata bahasa Inggris memiliki variasi dalam struktur dan penggunaannya, itu tergantung tradisi yang digunakan oleh suatu negara yang dipengaruhi oleh bahasa asli dari negara tersebut. Secara umum, tata bahasa yang dipedomani adalah tata bahasa Inggris Amerika dan Inggris Britania Raya (British). Sistem waktu dalam bahasa Inggris disebut tense. Tense terbagi menjadi tiga yakni:

- Past tense (waktu lampau)
- Present tense (waktu kini)
- Future tense (waktu akan datang)

Dalam satu tense masing-masing terbagi menjadi empat yakni: Simple, Continuous, Perfect, dan Perfect Continuous. Di Indonesia dikenal 16 sistem waktu (tenses) bahasa Inggris meski secara umum hanya 12 tenses. Tenses

secara sederhana dapat diartikan sebagai pola kalimat yang berubah menurut waktu merujuk pada masa lalu (Past), masa sekarang (Present) dan masa depan (Future). Dengan memahami tenses bahasa Inggris kita bisa menyusun kalimat dengan lebih baik meski tidak harus selalu kita gunakan. Berikut 16 sistem waktu bahasa Inggris yang akan kita pelajari dalam aplikasi ini:

1. Present Tenses

- (a) Simple Present Tenses (digunakan untuk menerangkan kejadian yang biasa atau sering terjadi setiap saat).
- (b) Present Continuous Tenses (digunakan untuk menyatakan peristiwa yang sedang terjadi saat pembicaraan berlangsung).
- (c) Present Perfect Tenses (digunakan untuk menerangkan peristiwa yang telah terjadi saat sekarang tetapi tidak diketahui dengan pasti waktunya).
- (d) Present Perfect Continuous Tenses (digunakan untuk menerangkan peristiwa yang telah terjadi dan masih sedang berlangsung)

2. Past Tenses

- (a) Simple Past Tenses (digunakan untuk menerangkan peristiwa yang terjadi beberapa saat yang lalu atau pada masa lampau).
- (b) Past Continuous Tenses (digunakan untuk menerangkan peristiwa yang sedang terjadi pada waktu lampau).
- (c) Past Perfect Tenses (digunakan untuk menerangkan peristiwa yang terjadi pada waktu lampau).
- (d) Past Perfect Continuous Tenses (digunakan untuk menerangkan hal yang sudah terjadi dan masih berlangsung pada waktu lampau)

3. Future Tenses

- (a) Simple Future Tenses (digunakan untuk menerangkan peristiwa yang akan terjadi atau akan dilakukan).
- (b) Future Continuous Tenses (digunakan untuk menerangkan hal yang sedang terjadi pada waktu yang akan datang).

(c) Future Perfect Tenses (digunakan untuk menerangkan hal yang akan terjadi atau dilakukan pada waktu yang akan datang).

(d) Future Perfect Continuous Tenses (digunakan untuk menerangkan peristiwa yang akan sudah dan masih berlangsung pada waktu yang akan datang)

4. Past Future Tenses

- (a) Simple Past Future Tenses (digunakan untuk menerangkan hal yang akan terjadi atau dilakukan pada akan datang diwaktu lampau).
- (b) Past Future Continuous Tenses (digunakan untuk menerangkan hal yang akan sedang terjadi pada waktu lampau).
- (c) Past Future Perfect Tenses (digunakan untuk menerangkan hal yang akan sudah selesai diwaktu lampau).
- (d) Past Future Perfect Continuous Tenses (digunakan untuk menerangkan hal yang akan telah sedang terjadi pada waktu lampau) [1].

Rumusan Masalah

Dalam penulisan ini penulis merencanakan membuat aplikasi untuk telepon seluler berbasis sistem operasi Android untuk pembelajaran Tenses dan Passive Voice dalam bahasa Inggris yang dapat dikenalkan [2]. Selain itu, didalam setiap tenses tersebut terdapat contoh dan latihan soal. Dalam aplikasi ini terdapat menu utama, dimana menu utama ini memiliki menu yang terdiri dari menu mulai, tentang dan bantuan.

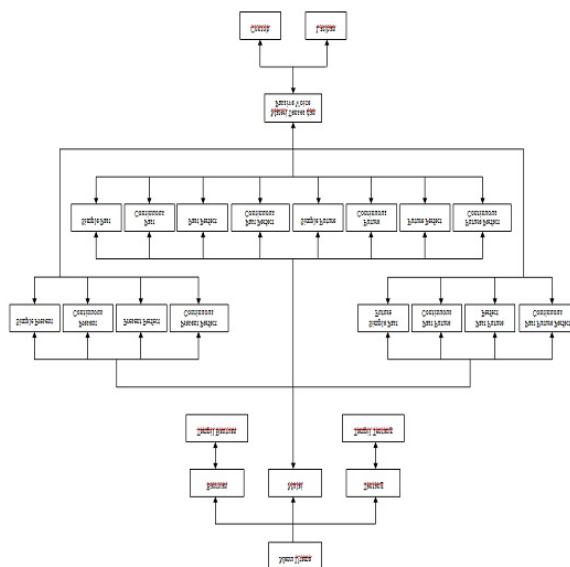
Menu mulai memiliki listview yang terdiri dari 16 tenses yang jika pengguna aplikasi memilih salah satu list tersebut, maka akan tampil materi tentang tenses dan passive voice yang dipilih serta tombol contoh dan latihan dibagian bawahnya. Jika pengguna menekan tombol contoh, maka akan tampil contoh tenses dan passive voicenya. Dan jika pengguna menekan tombol latihan, maka akan tampil satu buah soal essay yang berkaitan tentang tenses dan passive voice yang dipilih.

Pilihan tentang pada menu utama berisi informasi tentang pembuatan aplikasi dan pilihan bantuan pada menu utama berisi instruksi-

instruksi agar para pengguna lebih mudah dalam menggunakan aplikasi ini. Hal yang dilakukan penulis pertama kali adalah memilih bahasa pemrograman yang akan digunakan dan pokok bahasan atau materi-materi yang akan digunakan. Kemudian penulis melakukan pengumpulan data-data baik berupa materi, soal, gambar ataupun artikel dari internet yang berkaitan dengan aplikasi.

Gambar 1 adalah merupakan struktur program dari aplikasi pembelajaran tenses dan passive voice. Struktur Program yang digunakan dalam aplikasi ini adalah struktur composite.

Struktur program composite sering disebut struktur program campuran, yaitu merupakan sebuah struktur navigasi bebas yang merupakan gabungan dari ketiga struktur yang ada yaitu struktur linier, struktur hirarki dan struktur non linier. Struktur navigasi ini biasa digunakan dalam pembuatan multimedia karena dapat memberikan keinteraksian yang lebih tinggi.



Gambar 1: Struktur Navigasi

Penjelasan dari Struktur Navigasi:

1. Pada saat menjalankan aplikasi ini, tampilan pertama adalah halaman menu utama, pada menu utama ini terdapat pilihan mulai, tentang dan bantuan.
2. Bila pengguna memilih menu mulai maka akan muncul 16 list tenses yang berisi materi tenses dan passive voice. Saat pengguna memilih simple present, pengguna

akan memasuki tampilan materi tenses dan passive voice tentang simple present dan juga terdapat pilihan contoh dan latihannya. Begitu pula jika pengguna memilih tenses yang lainnya.

3. Bila pengguna memilih bantuan, maka akan tampil halaman panduan penggunaan aplikasi. Sehingga pengguna dapat menggunakan aplikasi ini dengan lebih mudah.
4. Setelah itu bila pengguna memilih menu tentang, maka pengguna akan memasuki tampilan yang berisikan informasi seputar aplikasi dan tujuan pembuatan aplikasi ini. Diagram alur (flowchart) Flowchart merupakan tahapan sistematis tentang suatu proses dan logika untuk mempermudah menggambarkan alur-alur dari suatu aplikasi. Proses dari sebuah aplikasi yang telah dibuat dapat dilihat melalui flowchart aplikasi Perancangan Tampilan Perancangan tampilan merupakan bagian kedua dari tahap pembuatan aplikasi ini. Perancangan ini sangat penting karena dengan membuat sebuah perancangan tampilan, pembuat aplikasi tidak akan mengalami kesulitan untuk membuat desain interface dari aplikasi yang dibuatnya.

Diagram Alur (Flowchart)

Flowchart merupakan tahapan sistematis tentang suatu proses dan logika untuk mempermudah menggambarkan alur-alur dari suatu aplikasi. Proses dari sebuah aplikasi yang telah dibuat dapat dilihat melalui flowchart aplikasi, lihat gambar 2.

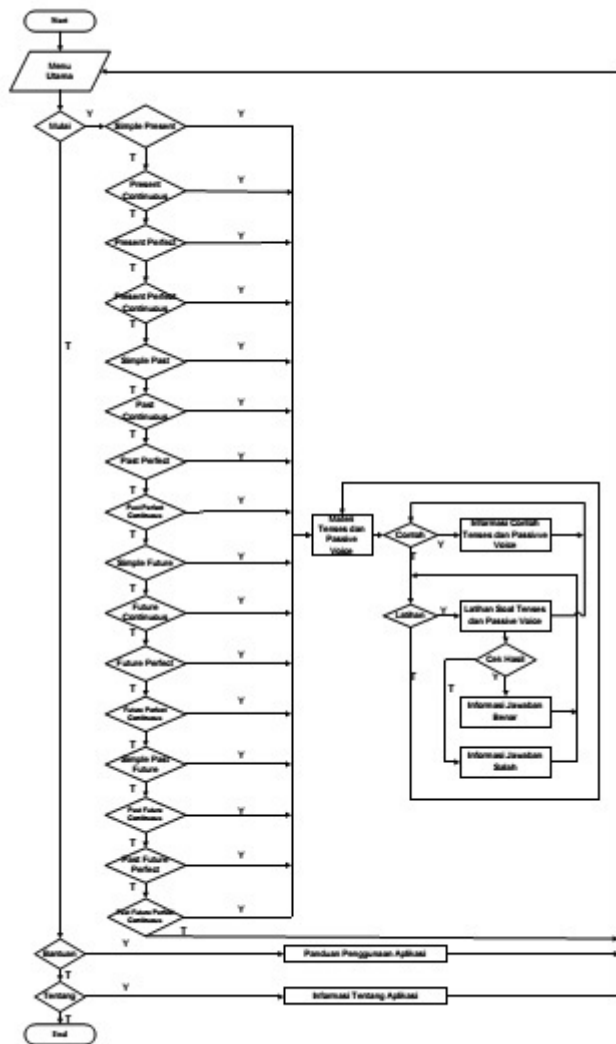
Perancangan Tampilan

Perancangan tampilan merupakan bagian kedua dari tahap pembuatan aplikasi ini. Perancangan ini sangat penting karena dengan membuat sebuah perancangan tampilan, pembuat aplikasi tidak akan mengalami kesulitan untuk membuat desain interface dari aplikasi yang dibuatnya.

Tampilan Menu Utama

Pada tampilan ini (gambar 3), terdapat title, nama aplikasi dan tiga buah button. Title pada halaman ini adalah ENGLISH FUN (title me-

nunjukkan nama label dari class yang aktif). Sedangkan button terdiri dari button mulai, button tentang, dan button bantuan. Setiap button yang ditekan akan masuk ke dalam sebuah class dan tampilan halaman baru.



Gambar 2: Diagram Alur

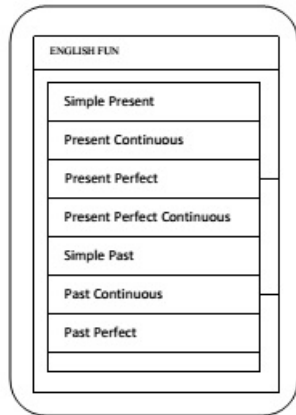
Pada tampilan gambar 4, terdapat title, 16 buah listview tenses dan layar yang bisa digeser ke atas dan ke bawah. Title pada halaman ini adalah ENGLISH FUN. Setiap list akan menampilkan materi tentang tenses sesuai dengan tensesnya serta terdapat pilihan contoh dan latihan.

Pada tampilan ini (gambar 5), terdapat title sesuai dengan tenses yang dipilih (Simple Present), 3 textview dan 2 button. Textview berikut berisi tulisan mengenai pengertian, rumus tenses dan rumus passive voice.

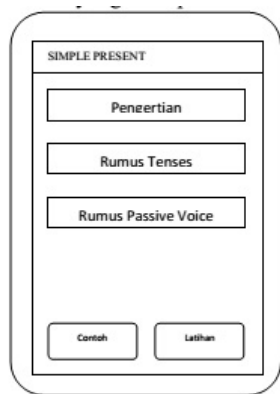


Gambar 3: Tampilan Utama

Button contoh berisi contoh-contoh tenses dan passive voice sesuai dengan tenses awal yang dipilih(Simple Present). Jika pengguna memilih button contoh maka akan masuk ke dalam class contoh simple present dan menampilkan halaman contoh simple present.



Gambar 4: Tampilan Tenses



Gambar 5: Tampilan Pilihan Tenses

Perancangan Database

Database pada aplikasi pembelajaran tenses dan passive voice ini sangat penting karena dalam penyimpanan soal-soal latihan tenses dan passive voice sangat dibutuhkan database untuk menyimpan seluruh soal dan jawaban dari soal tersebut. Tabel 1 ini digunakan untuk menyimpan semua macam soal latihan tenses dan passive voice.

Perancangan Tabel Android

Tabel android_metadata (tabel 2) ini digunakan untuk mengcopy file yang berisi SQL dari folder assets dan ke dalam database android.

Tabel 1: Tabel soal latihan tenses

FIELD	TYPE	KETERANGAN
_id	INTEGER PRIMARY KEY	Id untuk soal
soal	TEXT	Untuk setiap Soal
Jawaban	TEXT	Untuk Jawaban

Tabel 2: Tabel Anroid Metadata

FIELD	TYPE	KETERANGAN
locale	TEXT	Untuk menghubungkan database sqlite ke android

Implementasi

Pembuatan Database

Sebelum membuat aplikasi android, yang pertama dilakukan membuat database dengan isi dan tabelnya hanya untuk sementara atau sample. Pada pembuatan aplikasi ini digunakan sqlite untuk mengcopy databasanya kedalam database android sebenarnya. Langkah awalnya buka sqllitenya. Pilih new untuk membuat file database baru.

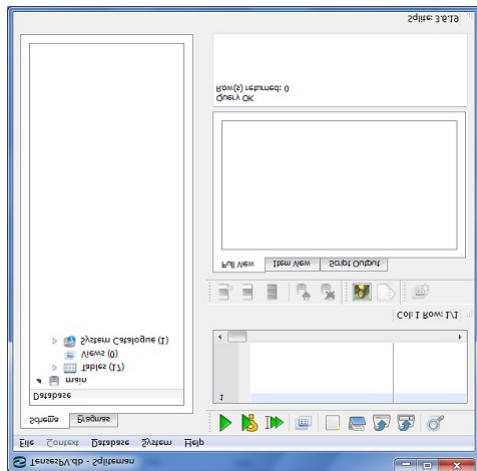
Setelah membuat databasanya. Sqlite ini sudah userinterface sehingga penulis tidak perlu menggunakan kodingan untuk membuat dan mengisi table. Cukup mengisi field-fieldnya dan mengisi recordnya secara manual. Setelah databasanya selesai file yang berekstensi .db itu harus di taruh dalam folder assets pada project android [4].

Pembuatan Project Aplikasi

Aplikasi ini dibuat dengan menggunakan Eclips Indigo dengan bahasa pemrograman Java untuk merancang dan membuat aplikasi untuk sistem operasi Android 3 Langkah pertama untuk membuat aplikasi adalah membuat file project baru dengan nama TENSEPV. Caranya dengan memilih menu File, New, pilih Project dan pilih Android Project maka akan masuk ke pengaturan project.

Pada pengaturan project, isi nama project, tentukan lokasi workspace yang akan digunakan, workspace adalah direktori yang menyimpan file project yang akan dibuat menggunakan Eclipse, biasanya workspace ditentukan secara default pada saat membuka Eclips. Tentukan build target, yaitu pengaturan versi be-

rapa yang ingin digunakan pada sistem operasi android , penulis memilih versi 2.2. lalu atur properties project, isi nama aplikasi yang nantinya akan keluar pada icon aplikasi yang digunakan , isi nama package minimal dua direktori, setiap nama direktori dipisahkan dengan tanda titik (.), kemudian buat nama activity, nama activity adalah yang akan muncul pada source programnya. Terakhir, tentukan versi minimal SDK, untuk Android versi 2.2 minimalnya adalah versi 8.



Gambar 6: Tampilan Sqliteman

Setelah project berhasil dibuat, tahap selanjutnya adalah membuat kelas-kelas java dan xml yang akan digunakan, serta menuliskan listing program aplikasi. Listing program adalah bagian utama untuk semua aplikasi, ini dibutuhkan agar aplikasi dapat berjalan dengan baik dan juga berguna untuk membuat agar tampilan aplikasi lebih menarik untuk dilihat.



Gambar 7: Tampilan Menu Utama

Tampilan Halaman Utama

Gambar 7 adalah tampilan menu utama. Tampilan menu utama adalah perpaduan dari layout main.xml dan listing program yang ada pada menu.java. Layout main.xml dapat dipanggil dengan cara: **setContent View(R.layout.main)**.

Pada menu utama ini pengguna dapat memilih salah satu dari 3 pilihan yang terdiri dari Mulai, Tentang dan Bantuan. Setiap pilihan yang dipilih, maka activity akan berjalan sesuai dengan pilihan yang dipilih. Listing program untuk menjalankan activity adalah sebagai berikut:

```
public void btnmulai (View v){ Intent n
=new Intent(this, ListTenses.class);
startActivity(n); }
public void btntentang (View v){ Intent n
=new Intent(this, Tentang.class);
startActivity(n); }
public void btnbantuan (View v){ Intent n
=new Intent(this, Bantuan.class);
startActivity(n);}
```

Gambar di atas adalah tampilan listview tenses yang menggunakan listtenses.xml. Layout listtenses.xml dapat dipanggil dengan cara: **setContent View(R.layout.listtenses)**;

Pada tampilan ini terdapat 16 listview tenses. Setiap listview tenses akan masuk kedalam tensesnya masing-masing. Semua listview tenses yang dipilih akan masuk ke activity baru sesuai dengan nama listviewnya. Listing program pada listtenses.java untuk masuk ke activity barunya adalah sebagai berikut:

```
public class
ListTenses extends Activity
{ String[] mpresent
= new String[]
{ "Simple Present",
"Present Continuous",
"Present Perfect",
"Present Perfect Continuous",
"Simple Past",
"Past Continuous",
"Past Perfect",
"Past Perfect Continuous",
"Simple Future",
"Future Continuous",
"Future Perfect",
"Future Perfect Continuous",
"Simple Past Future",
"Past Future Continuous",
"Past Future Perfect",
```

"Past Future Perfect Continuous"};



Gambar 8: Tampilan Listview Tenses

Penjelasan listing program diatas adalah sebuah string listview dengan nama mpresent yang berisi "Simple Present", "Present Continuous", "Present Perfect", "Present Perfect Continuous", "Simple Past", "Past Continuous", "Past Perfect", "Past Perfect Continuous", "Simple Future", "Future Continuous", "Future Perfect", "Future Perfect Continuous", "Simple Past Future", "Past Future Continuous", "Past Future Perfect", "Past Future Perfect Continuous".

Isi listview tersebut akan muncul ditampilan listview dengan cara berurutan sesuai dengan string yang ditulis terlebih dahulu. Sedangkan untuk pemanggilan activity, digunakan kondisi sebagai berikut:

```

if(mpresent[position].equals
("Simple Present"))
{Intent i = new Intent
(ListTenses.this, SPresent.class);
startActivity(i);} else
if (mpresent[position].
equals("Present Continuous"))
{Intent i = new Intent
(ListTenses.this, PresentCon.class);
startActivity(i);} else
if (mpresent[position].
equals("Present Perfect"))
{ Intent i = new Intent
(ListTenses.this, PresentPer.class);
startActivity(i);} else
if (mpresent[position].
equals("Present Perfect Continuous"))
{ Intent i = new Intent

```

```

(ListTenses.this, PresentPerCon.class);
startActivity(i);}
if(mpresent[position].
equals("Simple Past"))
{ Intent i = new Intent
(ListTenses.this, SPast.class);
startActivity(i);} else
if (mpresent[position].
equals("Past Continuous"))
{ Intent i = new Intent
(ListTenses.this, PastCon.class);
startActivity(i);} else
if (mpresent[position].
equals("Past Perfect"))
{ Intent i = new Intent
(ListTenses.this, PastPer.class);
startActivity(i);} else
if (mpresent[position].
equals("Past Perfect Continuous"))
{ Intent i = new Intent
(ListTenses.this, PastPerCon.class);
startActivity(i);}
if(mpresent[position].
equals("Simple Future"))
{ Intent i = new Intent
(ListTenses.this, SFuture.class);
startActivity(i);} else
if (mpresent[position].
equals("Future Continuous"))
{ Intent i = new Intent
(ListTenses.this, FutureCon.class);
startActivity(i);} else
if (mpresent[position].
equals("Future Perfect"))
{ Intent i = new Intent
(ListTenses.this, FuturePer.class);
startActivity(i);} else
if (mpresent[position].
equals("Future Perfect Continuous"))
{ Intent i = new Intent
(ListTenses.this, FuturePerCon.class);
startActivity(i);}
if(mpresent[position].
equals("Simple Past Future"))
{ Intent i = new Intent
(ListTenses.this, SPastFuture.class);
startActivity(i);} else
if (mpresent[position].
equals("Past Future Continuous"))
{ Intent i = new Intent
(ListTenses.this, PFutureCon.class);
startActivity(i);} else
if (mpresent[position].

```

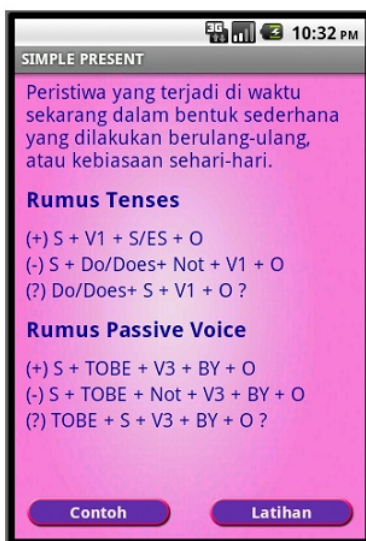
```

equals("Past Future Perfect"))
{Intent i = new Intent
(ListTenses.this, PFuturePer.class);
startActivity(i);} else
if (mpresent[position].
equals("Past Future Perfect Continuous"))
{ Intent i = new Intent
(ListTenses.this, PFuturePerCon.class);
startActivity(i);}
    
```

Penjelasan listing programnya adalah jika isi string yang ada pada mpresent sama dengan nama tensesnya, maka program akan menjalankan activity selanjutnya. Misal, jika isi string di mpresent sama dengan Simple Present, maka program akan memanggil activity SPresent.class untuk activity selanjutnya.

Tampilan Materi Tenses

Gambar 8 adalah tampilan materi tenses (Simple Present) yang menggunakan spresent.xml. Layout spresent.xml dapat dipanggil dengan cara: setContentView(R.layout.spresent);



Gambar 9: Tampilan Materi tenses

Pada gambar 9 terdapat Pengertian simple present, rumus tenses dan rumus passive voice. Dan pada tampilan ini juga terdapat tombol contoh dan latihan tensesnya. Pada pengertian, rumus tenses dan rumus passive voice, penulis menggunakan string.xml, dimana string.xml berguna untuk menuliskan isi text yang panjang. String.xml terdapat di directory values. Isi dari dari string.xml dipanggil dilayout yang diinginkan dibagian textview. Misalnya android:text="@String/SPresentKet" yang maksudnya adalah program akan mengambil

text dari string.xml dengan nama string SpresentKet. Sedangkan pada tombol Contoh dan Latihan menggunakan listing program sebagai berikut :

```

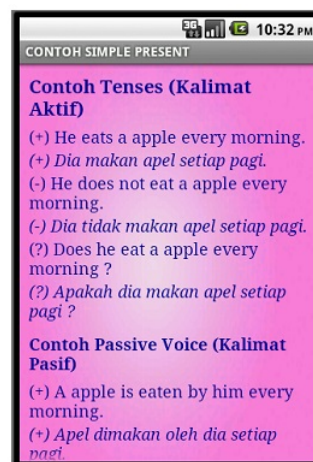
public void btnContoh (View v)
{ Intent n =
new Intent (this, CSPresent.class);
startActivity(n);}
public void btnlatihan (View v)
{ Intent n =
new Intent (this, LPresent.class);
startActivity(n);}
    
```

Penjelasan listing programnya adalah sebuah pendefinisian intent yang berfungsi untuk pindah activity dan menuju ke activity yang baru atau activity selanjutnya. Jika pengguna memilih Contoh, maka activity baru akan berjalan yaitu akan menuju ke CSPresent.class dan jika pengguna memilih Latihan, maka activity selanjutnya akan menuju ke LPresent.class.

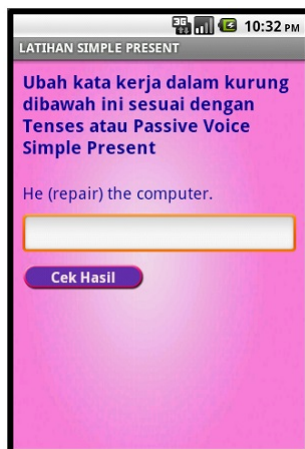
Tampilan Contoh Tenses

Tampilan contoh tenses di atas (Simple Present) menggunakan cspresent.xml. Layout cspresent.xml yang dapat dipanggil dengan cara: setContentView(R.layout.cspresent);

Pada gambar 10 hanya terdapat contoh tenses dan contoh passive voice yang berupa text-text. Pada contoh tenses dan passive voice terdapat contoh kalimat positif, kalimat negatif dan kalimat tanya yang masing-masing ada artinya. Semua text-text tersebut disimpan di dalam string.xml.



Gambar 10: Tampilan Contoh Tenses dan Bentuk Pasifnya



Gambar 11: Tampilan Latihan Tenses

Tampilan Latihan Tenses

Tampilan gambar 11 adalah tampilan Latihan tenses (Simple Present) yang menggunakan `lspresent.xml`. Layout `lspresent.xml` dapat dipanggil dengan cara : `setContentView(R.layout.lspresent);`

Pada tampilan ini hanya terdapat 2 textview, yang pertama untuk perintah soal dan yang kedua untuk soalnya. 1 edittext untuk tempat menjawab soal dan 1 button cek hasil. Dalam pembuatan tampilan ini pertama kita membuat method `view()` untuk menampilkan soal dari database yang berisi kode program seperti dibawah:

```

SQLiteDatabase db =
dbHelper.getReadableDatabase();
try { cursor = db.rawQuery
( "SELECT *
FROM lspresent order by random()
limit 1 ", null);
int kolom0 =
cursor.getColumnIndex("Soal");
int kolom1 =
cursor.getColumnIndex("Jawaban");
cursor.moveToFirst(); String text =
cursor.getString(kolom0);
jawab =
cursor.getString(kolom1);
txtview.setText(text); }
catch (Exception e) {

```

Penjelasan listing program diatas untuk menampilkan soal dari database secara acak dan mengambil serta memindahkan nilai index dari soal dengan cursor. Selain itu, variable string `text` untuk mengambil string `text` soal

dan jawab untuk mengambil nilai string jawaban. Kemudian buat method `btnjawab()` untuk mengetahui nilai string yang dimasukkan pengguna apakah sama dengan jawaban pada database yang berisi kode program seperti dibawah ini:

```

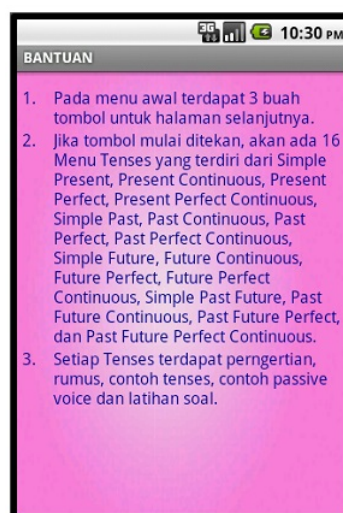
String jwb = edit.getText().toString();
if (jwb.equals(jawab))
{ Toast.makeText(LPresent.this,
"Jawaban Benar",
Toast.LENGTH_SHORT) .show(); }
else { Toast.makeText(LPresent.this,
"Jawaban salah",
Toast.LENGTH_SHORT) .show();} }

```

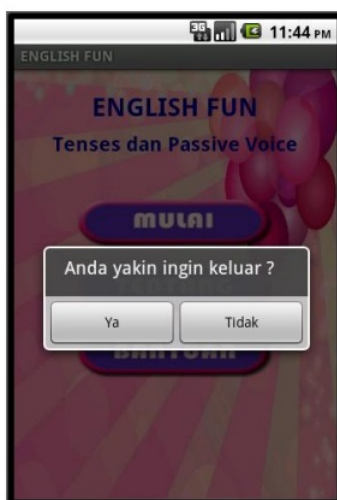
Penjelasan listing program diatas adalah untuk mengambil nilai yang didapat dari `edittext` (string yang dimasukkan pengguna) kedalam nama variabel string `jwb`. Setelah itu, dibuat kondisi dimana jika nilai string yang dimasukkan pengguna sama dengan jawaban yang ada didatabase, maka program akan menampilkan tulisan jawaban benar pada toast dan jika nilai string yang dimasukkan pengguna tidak sama dengan jawaban yang ada didatabase, maka program akan menampilkan tulisan jawaban salah pada toast.

Tampilan Bantuan

Gambar 12 adalah tampilan untuk bantuan tentang cara menggunakan aplikasi dan penjelasan fungsi tombol-tombol yang ada pada aplikasi ini. Untuk memanggil layout bantuan `bantuan.xml` dapat dilakukan dengan cara: `setContentView(R.layout.bantuan);` Isi dari layout ini hanya terdiri dari textview yang disusun secara relative.



Gambar 12: Tampilan Bantuan



Gambar 13: Tampilan Keluar

Tampilan Keluar

Gambar 13 adalah tampilan dari dialog yang ada pada class menu.java, dialog ini aktif ketika pengguna menekan tombol back pada fungsi android, jika pengguna memilih ya, maka pengguna akan keluar menuju tampilan home dari device, karena listing program dibawah ini:

```
AlertDialog.Builder alertKeluar = new
AlertDialog.Builder(menu.this); alertKeluar
.setMessage("Anda yakin ingin keluar ?")
.setCancelable(false) .setPositiveButton("Ya",
new AlertDialog.OnClickListener() { public
void onClick(DialogInterface arg0, int arg1)
{ finish();}}).
```

AlertDialog akan muncul secara otomatis saat pengguna menekan tombol keluar. AlertDialog adalah kotak kecil bertuliskan "Anda yakin ingin keluar ?" dan disertai 2 tombol yang bernilai positif dan negative. Jika memilih tidak, pengguna akan tetap berada di tampilan utama. Berikut ini listing programnya:

```
setNegativeButton("Tidak", new Alert-
Dialog.OnClickListener() { public void
onClick(DialogInterface dialog,int which)
{ dialog.cancel(); } }); AlertDialog a =
alertKeluar.create(); a.show(); }
```

Penjelasan listing program diatas adalah pada menu.java terdapat di onBackPressed, yaitu fungsi back pada menu android. Pada onBackPressed tersebut didefinisikan sebuah objek AlertDialog.Builder dengan nama alertKeluar, objek tersebut menggunakan class menu, karena ada

keyword this, yang artinya objek yang langsung digunakan tanpa didahului proses instansiasi. Setelah itu merupakan listing program untuk mendefinisikan fungsi dari tombol-tombol yang terdapat pada objek AlertDialog

Penutup

Aplikasi ini dapat membantu pengguna dalam belajar tenses dan passive voice serta dapat mempermudah pengguna dalam belajar tanpa batasan waktu dan tempat penggunaannya, dikarenakan aplikasi ini diimplementasikan pada telepon seluler yang berbasis Android. Selain itu, pengguna juga dapat mempelajari tenses dan passive voice melalui contoh dan latihan yang ada pada aplikasi ini.

Penulis menyadari bahwa masih terdapat kekurangan dalam aplikasi ini, untuk itu penulis hendak memberikan saran bagi pengembangan dan penyempurnaan aplikasi ini. Pada Aplikasi ini, hanya membahas tentang pembelajaran tenses dan passive voice yang pada setiap tensesnya hanya terdapat 1 contoh tenses (kalimat aktif) dan 1 contoh passive voice (kalimat pasif) serta 1 latihan tenses atau passive voice saja. Saran penulis untuk pengembangan selanjutnya, aplikasi pembelajaran tenses dan passive voice ini diharapkan dapat menampilkan contoh dan latihan yang lebih banyak lagi. Selain itu juga bisa ditambahkan untuk pembelajaran grammar secara menyeluruh.

Daftar Pustaka

- [1] Agung Prakoso, "Aplikasi Penngenanlan Bahasa Inggris untuk SD Kelas 3 Menggunakan Android versi 2.2", 2012.
- [2] Andi, "Step by Step Menjadi Programmer Android", Semarang:Wahana Komputer, 2013.
- [3] Ir. Yuniar Supardi, "Semua Bisa Menjadi Programmer Android Basic", Jakarta:Elex Media Komputindo, 2011.
- [4] Nazruddin Safaat H, "Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android", Bandung:Informatika, 2011.