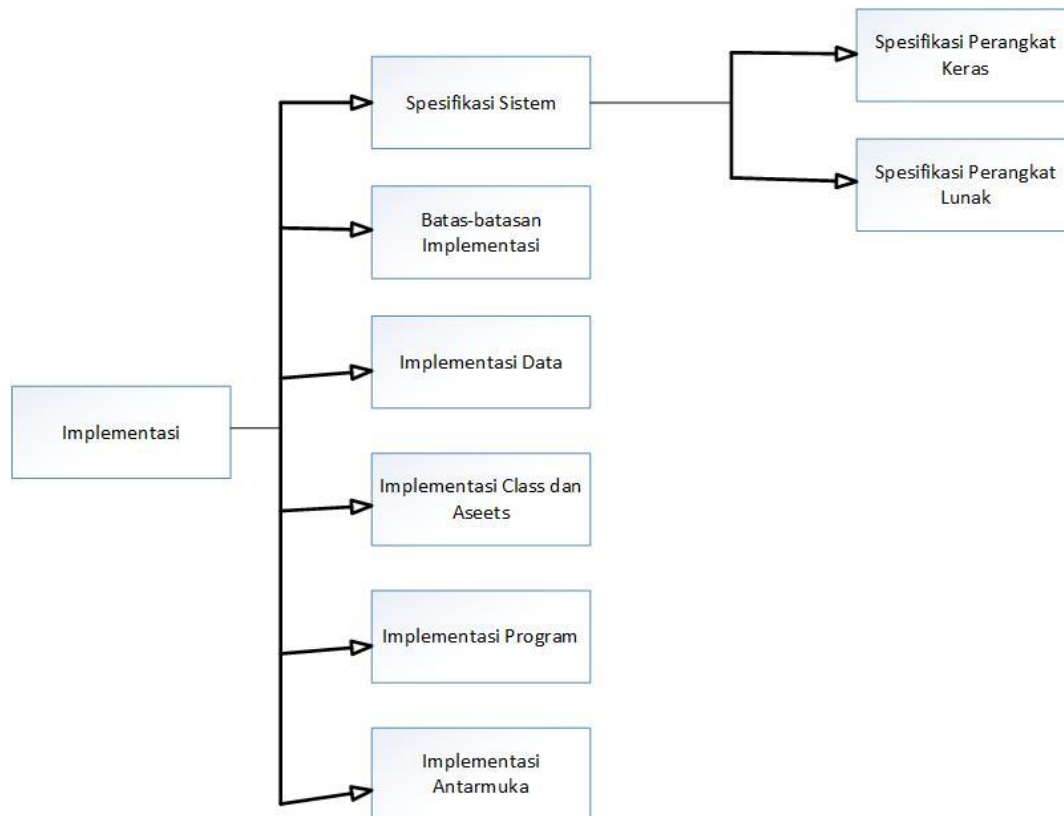


BAB 5 IMPLEMENTASI

Pada bab ini akan menjelaskan implementasi Aplikasi “**DekatTuli**” berbasis android dengan metode *speech recognition* berdasarkan analisis kebutuhan dan perancangan yang telah dilakukan sebelumnya. Struktur dan langkah – langkah implementasi aplikasi yang dibangun dijelaskan pada gambar 5.1.



Gambar 0.1 Diagram Pohon Implementasi

Berdasarkan diagram pada gambar 5.1 dapat dijelaskan bahwa dalam tahap implementasi dibutuhkan spesifikasi sistem yang terbagi menjadi spesifikasi perangkat keras dan spesifikasi perangkat lunak, selain itu, terdapat juga penjelasan mengenai batasanbatasan implementasi, implementasi basis data, implementasi *class* dan *assets*, implementasi kode program dan implementasi antarmuka.

5.1 Spesifikasi Sistem

Berdasarkan perancangan arsitektur perangkat bergerak yang telah dijelaskan pada Gambar4.4 menjadi acuan dalam implementasi dan berfungsi sesuai dengan kebutuhan analisis. Spesifikasi sistem di implementasi menjadi dua jenis, yaitu spesifikasi perangkat keras dan spesifikasi perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam implementasi Aplikasi komunikasi Belajar Bahasa Isyarat Indonesia ditunjukkan pada Tabel 5.1

Tabel 0.1 Spesifikasi Perangkat Keras

Komponen	Spesifikasi
<i>System model</i>	Toshiba C50-b
<i>Processor</i>	Intel(R) Core (TM) i3-3217U CPU @180GHz 1.80GHz
<i>Memory</i>	4 GB RAM
<i>Display</i>	Intel® HD Graphic
<i>System tiye</i>	64 bit Operting System, x64based processor

Untuk proses instalasi dan pengujian berupa perangkat Android perangkat bergerak dengan spesifikasi yang ditunjukkan pada Tabel 5.2

Tabel 0.2 Spesifikasi Perangkat Lunak

Komponen	Spesifikasi
<i>System model</i>	Lenovo A3300
<i>OS</i> <i>Chipset</i> <i>CPU</i> <i>GPU</i>	Android OS, v4.2.2 (Jelly Bean) Mediatek MT8382M Quad-core 1.3 GHz Cortex-A7 Mali-400MP2
<i>Memory</i>	16 GB, 1 GB RAM
<i>Display</i>	600 x 1024 pixels (~170 ppi pixel density)
Fisik	194 x 120 x 11 mm (7.64 x 4.72 x 0.43 in)

5.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam implementasi Aplikasi komunikasi Belajar Bahasa Isyarat Indonesia ditunjukkan pada Tabel 5.3

Tabel 0.3 Spesifikasi Perangkat Lunak dan Komputer

Komponen	Spesifikasi
Sistem Operasi	Windows 8 pro
Bahasa Pemrograman	Java dan XML
Tools Development	Android Studio, SQLite Studio, dan Adobe Premiere

5.2 Batasan-batasan Implementasi

Aplikasi perangkat bergerak memiliki beberapa batasan dalam proses pengerjaan pada tahap implementasi, yaitu:

1. Aplikasi Papan Komunikasi berbasis Tablet Android untuk Belajar Bahasa Isyarat Indonesia yang diimplementasikan menggunakan sistem operasi Android minimal versi 4.0 Jelly Bean.
2. Aplikasi perangkat bergerak diimplementasikan menggunakan *tool development* Android Studio dan Bahasa pemrograman Java dan xml.
3. Penyimpan data menggunakan *database* SQLite disimpan secara *local storage* pada perangkat bergerak.

5.3 Implementasi Class dan Assets pada File Program

Berikut adalah setiap *class* dan *assets* yang telah dirancang pada tahap perancangan class diagram dalam Gambar 4.7 untuk aplikasi perangkat bergerak direalisasikan pada beberapa berkas file program yang dijelaskan pada Tabel 5.6.

Tabel 0.4 Berkas Class dan Assets Layout pada Aplikasi Perangkat Bergerak

No	Package	Nama Class	Nama Layout
1.	com.example.blackjack.communication.activity	MainActivity	activity_main
2.	com.example.blackjack.communication.activity	Communication Activity	activity_com_men u
3.	com.example.blackjack.communication.activity	KataTanyaActivity	activity_com_katat anya
4.	com.example.blackjack.communication.activity	PercakapanActivity	activity_com_seha rihari
5.	com.example.blackjack.communication.activity	SpokActivity	activity_com_spok

5.4 Implementasi Kode Program

Aplikasi perangkat bergerak ini memiliki beberapa fungsi utama yang ada di beberapa class pada *package* courier new. Implementasi kode program ini dibuat berdasarkan pada perancangan class diagram dalam Gambar 4.14 bagian *class MainActivity* yang digunakan untuk menampilkan berita pada halaman beranda.

5.4.1 Implementasi Kode Program Halaman Utama

Implementasi Kode Program Halaman Utama class *CommunicationActivity* yang digunakan untuk menampilkan menu utama pada aplikasi “**DekatTuli**” yang ditunjukkan pada Kode 5.1 sebagai berikut:

```

1 package com.example.Communication.dekattuli.activity;
2
3 import android.content.Intent;
4 import android.os.Bundle;
5 import android.support.v7.app.AppCompatActivity;
5 import android.view.View;
6 import android.widget.Button;
7
8 import com.example.Hasan.DekatTuli.R;
9
10 public class CommunicationActivity extends AppCompatActivity {
11
12     private Button btnTanya, btnPercakapan, btnSpok;
13
14     public void assignView(){
15         btnTanya = (Button)findViewById(R.id.btnTanya);
16         btnPercakapan = (Button)findViewById(R.id.btnPercakapan);
17         btnSpok = (Button)findViewById(R.id.btnSpok);
18     }
19
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         setContentView(R.layout.activity_com_menu);
24
25         assignView();
26
27         btnTanya.setOnClickListener(new View.OnClickListener() {
28             @Override
29             public void onClick(View v) {
30                 Intent tanya = new
31 Intent(CommunicationActivity.this, KataTanyaActivity.class);
32                 startActivity(tanya);
33             }
34         });
35
36         btnPercakapan.setOnClickListener(new View.OnClickListener()
37 {
38             @Override
39             public void onClick(View v) {
40                 Intent kosakata = new
41 Intent(CommunicationActivity.this, PercakapanActivity.class);
42                 startActivity(kosakata);
43             }
44         });
45
46         btnSpok.setOnClickListener(new View.OnClickListener() {
47             @Override
48             public void onClick(View v) {
49                 Intent kosakata = new
50 Intent(CommunicationActivity.this, SpokActivity.class);
51                 startActivity(kosakata);
52             }
53         });
54     }
55 }

```

Kode 0.1 Implementasi Kode Program Utama Menu

Di bawah ini merupakan keterangan kode Program Halaman Utama:

Nomer	Kode Program	Keterangan
27	btnTanya.setOnClickListener(new View.OnClickListener())	Tombol Menu Kata Tanya

36	btnPercakapan.setOnClickListener(new View.OnClickListener())	Tombol Menu Percakapan Sehari-hari
46	btnSpok.setOnClickListener(new View.OnClickListener())	Tombol Menu SPOK

5.4.2 Implementasi Kode Program Komunikasi

Implementasi Kode Program komunikasi class CommunicationActivity yang digunakan untuk menampilkan menu halaman pembelajaran pada aplikasi “DekatTuli” yang ditunjukkan pada Kode 5.2 sebagai berikut:

```

1 package com.example.Communication.dekattuli.activity;
2
3 import android.content.Intent;
4 import android.os.Bundle;
5 import android.support.v7.app.AppCompatActivity;
5 import android.view.View;
6 import android.widget.Button;
7
8 import com.example.Communication.dekattuli.R;
9
10
11 public class CommunicationActivity extends AppCompatActivity {
12
13     private Button btnTanya, btnPercakapan, btnSpok;
14
15     public void assignView(){
16         btnTanya = (Button)findViewById(R.id.btnTanya);
17         btnPercakapan = (Button)findViewById(R.id.btnPercakapan);
18         btnSpok = (Button)findViewById(R.id.btnSpok);
19     }
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_com_menu);
25
26         assignView();
27
28         btnTanya.setOnClickListener(new View.OnClickListener() {
29             @Override
30             public void onClick(View v) {
31                 Intent tanya = new
32                 Intent(CommunicationActivity.this, KataTanyaActivity.class);
33                 startActivity(tanya);
34             }
35         });
36
37         btnPercakapan.setOnClickListener(new View.OnClickListener()
38     {
39             @Override
40             public void onClick(View v) {
41                 Intent kosakata = new
42                 Intent(CommunicationActivity.this, PercakapanActivity.class);
43                 startActivity(kosakata);
44             }
45         });
46
47         btnSpok.setOnClickListener(new View.OnClickListener() {
48             @Override
49             public void onClick(View v) {
50                 Intent kosakata = new
51                 Intent(CommunicationActivity.this, SpokActivity.class);
52                 startActivity(kosakata);
53         }

```

```

54         });
55     }
56 }

```

Kode 0.2 Implementasi Kode Program Komunikasi

Di bawah ini merupakan keterangan kode Program Komunikasi:

Nomer	Kode Program	Keterangan
28	<code>btnTanya.setOnClickListener(new View.OnClickListener())</code>	Tombol Menu Kata Tanya
37	<code>btnPercakapan.setOnClickListener(new View.OnClickListener())</code>	Tombol Menu Percakapan Sehari-hari
47	<code>btnSpok.setOnClickListener(new View.OnClickListener())</code>	Tombol Menu SPOK

5.4.3 Implementasi Kode Program Kata Tanya

Implementasi Kode Program komunikasi class `KataTanyaActivity` yang digunakan untuk menampilkan menu halaman pembelajaran pada aplikasi “**DekatTuli**” yang ditunjukkan pada Kode 5.3 sebagai berikut:

```

1 package com.example.KataTanyaActivity.DekatTuli.activity;
2
3 import android.app.ProgressDialog;
4 import android.content.Context;
5 import android.content.Intent;
6 import android.content.res.Resources;
7 import android.net.Uri;
8 import android.os.AsyncTask;
9 import android.os.Bundle;
10 import android.speech.tts.TextToSpeech;
11 import android.support.v7.app.AppCompatActivity;
12 import android.support.v7.widget.LinearLayoutManager;
13 import android.support.v7.widget.RecyclerView;
14 import android.util.Log;
15 import android.view.View;
16 import android.widget.Button;
17 import android.widget.Toast;
18 import android.widget.VideoView;
19
20 import com.example.donisaurus.DekatTuli.R;
21 import com.example.donisaurus.DekatTuli.adapter.KataTanyaAdapter;
22 import com.example.donisaurus.DekatTuli.adapter.KosaKataAdapter;
23 import com.example.donisaurus.DekatTuli.adapter.RecyclerTouchListener;
24 import com.example.donisaurus.DekatTuli.database.KosaKata;
25
26 import org.w3c.dom.Document;
27 import org.w3c.dom.Element;
28 import org.w3c.dom.Node;
29 import org.w3c.dom.NodeList;
30
31 import java.io.InputStream;
32 import java.util.ArrayList;
33 import java.util.Locale;
34
35 import javax.xml.parsers.DocumentBuilder;
36 import javax.xml.parsers.DocumentBuilderFactory;
37

```

```

38  /**
39   * Created by Donisaurus on 3/19/2017.
40   */
41
42  public class KataTanyaActivity extends AppCompatActivity
43  implements TextToSpeech.OnInitListener{
44      private RecyclerView rvKataTanya;
45      private VideoView vvKataTanya;
46      private TextToSpeech myTTS;
47      private int MY_DATA_CHECK_CODE = 0;
48
49      public void assignView(){
50          rvKataTanya =
51 (RecyclerView)findViewById(R.id.rvKataTanya);
52          vvKataTanya = (VideoView)findViewById(R.id.vvKataTanya);
53      }
54
55      @Override
56      protected void onCreate(Bundle savedInstanceState) {
57          super.onCreate(savedInstanceState);
58          setContentView(R.layout.activity_com_katatanya);
59
60          assignView();
61
62          //check for TTS data
63          Intent checkTTSIntent = new Intent();
64
65 checkTTSIntent.setAction(TextToSpeech.Engine.ACTION_CHECK_TTS_DATA
66 );
67          startActivityForResult(checkTTSIntent,
68 MY_DATA_CHECK_CODE);
69      }
70
71      //act on result of TTS data check
72      protected void onActivityResult(int requestCode, int
73 resultCode, Intent data) {
74
75          if (requestCode == MY_DATA_CHECK_CODE) {
76              if (resultCode ==
77 TextToSpeech.Engine.CHECK_VOICE_DATA_PASS) {
78                  //the user has the necessary data - create the TTS
79                  myTTS = new TextToSpeech(this, this);
80
81                  //Read XML from device
82                  KataTanyaActivity.ReadXml readXml = new
83 KataTanyaActivity.ReadXml(this, rvKataTanya, vvKataTanya, myTTS);
84                  readXml.execute();
85              }
86              else {
87                  //no data - install it now
88                  Intent installTTSIntent = new Intent();
89
90 installTTSIntent.setAction(TextToSpeech.Engine.ACTION_INSTALL_TTS_
91 DATA);
92                  startActivity(installTTSIntent);
93              }
94          }
95      }
96
97      @Override
98      public void onInit(int initStatus) {
99          Locale loc= new Locale("ind","IDN");
100         //check for successful instantiation
101         if (initStatus == TextToSpeech.SUCCESS) {
102             if(myTTS.isLanguageAvailable(new
103 Locale("ind","IDN"))==TextToSpeech.LANG_AVAILABLE)
104                 myTTS.setLanguage(loc);
105         }
106     }

```

```

102     else if (initStatus == TextToSpeech.ERROR) {
103         Toast.makeText(this, "Sorry! Text To Speech
104 failed...", Toast.LENGTH_LONG).show();
105     }
106 }
107
108     private class ReadXml extends AsyncTask<Void, Void, Void> {
109         Context context;
110         ProgressDialog progressDialog;
111         ArrayList<String> listKataTanya;
112         RecyclerView recyclerView;
113         VideoView vvKataTanya;
114         TextToSpeech myTTS;
115
116         public ReadXml(Context context, RecyclerView recyclerView,
117 VideoView vvKataTanya, TextToSpeech myTTS){
118             this.recyclerView = recyclerView;
119             this.context = context;
120             this.vvKataTanya = vvKataTanya;
121             this.myTTS = myTTS;
122             progressDialog = new ProgressDialog(context);
123             progressDialog.setMessage("Loading...");
124         }
125
126         @Override
127         protected void onPreExecute() {
128             progressDialog.show();
129             super.onPreExecute();
130         }
131
132         @Override
133         protected void onPostExecute(Void aVoid) {
134             super.onPostExecute(aVoid);
135             progressDialog.dismiss();
136
137             KataTanyaAdapter adapter = new
138 KataTanyaAdapter(context, listKataTanya, vvKataTanya, myTTS);
139 recyclerView.setLayoutManager(new
140 LinearLayoutManager(context));
141 recyclerView.setAdapter(adapter);
142
143 //         recyclerView.addOnItemTouchListener(new
144 9 RecyclerViewTouchListener(getApplicationContext(), recyclerView, new
145 ClickListener() {
146 //             @Override
147 //             public void onClick(View view, int position) {
148 //                 Resources res = getResources();
149 //                 int videoId =
150 res.getIdentifier(listKataTanya.get(position), "raw",
151 getPackageName());
152 //                 //
153 //                 //Set Uri Video
154 //                 Uri videoPath =
155 Uri.parse("android.resource://" + getPackageName() + "/" +
156 videoId);
157 //                 vvKataTanya.setVideoURI(videoPath);
158 //                 vvKataTanya.start();
159 //                 Log.d("tanya", videoPath + "");
160 //                 myTTS.speak("apa kabar Doni Putra Purbawa,
161 anak ganteng", TextToSpeech.QUEUE_FLUSH, null);
162 //             }
163 //
164 //             @Override
165 //             public void onLongClick(View view, int position)
166 {
167 //             }
168 //         });
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }

```



```

193     }
194
195     @Override
196     protected Void doInBackground(Void... params)
197     {
198         ProcessXml(GetData());
199         return null;
200     }
201
202     @Override
203     protected void onProgressUpdate(Void... text) {
204         //         finalResult.setText(text[0]);
205     }
206
207     private void ProcessXml(Document data) {
208         if (data != null){
209             listKataTanya = new ArrayList<>();
210             Element root = data.getDocumentElement();
211             Node tabelkatatanya =
root.getChildNodes().item(1);
212             NodeList items = tabelkatatanya.getChildNodes();
213
214             for (int i = 0; i < items.getLength(); i++){
215                 Node currentChild = items.item(i);
216                 if
(currentChild.getNodeName().equalsIgnoreCase("value")){
217                     Log.d("node",
218 currentChild.getTextContent().substring(5,
219 currentChild.getTextContent().length()) + "");
220
221 listKataTanya.add(currentChild.getTextContent());
222                 }
223             }
224             }else{
225                 Log.d("Root", "gagal / data null");
226             }
227         }
228
229     public Document GetData(){
230         try {
231             InputStream inputStream =
getResources().openRawResource(R.raw.katatanya);
232             DocumentBuilderFactory builderFactory =
DocumentBuilderFactory.newInstance();
233             DocumentBuilder builder =
builderFactory.newDocumentBuilder();
234             Document xmlDoc = builder.parse(inputStream);
235             return xmlDoc;
236         } catch (Exception e){
237             e.printStackTrace();
238             return null;
239         }
240     }
241 }
242 }
243 }
244
245
256

```

Kode 0.3 Implementasi Kode Program Kata Tanya

Di bawah ini merupakan keterangan kode Program Kata Tanya:

Nomer	Kode Program	Keterangan
-------	--------------	------------

80	<pre>KataTanyaActivity.ReadXml readXml = new KataTanyaActivity.ReadXml(this, rvKataTanya, vvKataTanya, myTTS); readXml.execute();</pre>	Kode membaca dari Data XML dari <i>device</i>
223	<pre>listKataTanya.add currentChild.getTextContent());</pre>	Kode <i>List</i> kata tanya

5.4.4 Implementasi Kode Program Percakapan Sehari – hari

Implementasi Kode Program komunikasi class PercakapanActivity yang digunakan untuk menampilkan menu halaman pembelajaran pada aplikasi “**DekatTuli**” yang ditunjukkan pada Kode 5.4 sebagai berikut:

```

1 package com.example.donisaurus.dekattuli.activity;
2
3 import android.app.ProgressDialog;
4 import android.content.Context;
5 import android.content.Intent;
5 import android.content.res.Resources;
6 import android.net.Uri;
7 import android.os.AsyncTask;
8 import android.os.Bundle;
9 import android.speech.tts.TextToSpeech;
10 import android.support.v7.app.AppCompatActivity;
11 import android.support.v7.widget.LinearLayoutManager;
12 import android.support.v7.widget.RecyclerView;
13 import android.util.Log;
14 import android.view.View;
15 import android.widget.Toast;
16 import android.widget.VideoView;
17
18 import com.example.donisaurus.DekatTuli.R;
19 import com.example.donisaurus.DekatTuli.adapter.KataTanyaAdapter;
20 import com.example.donisaurus.DekatTuli.adapter.PercakapanAdapter;
21 import
22 com.example.donisaurus.DekatTuli.adapter.RecyclerTouchListener;
23
24 import org.w3c.dom.Document;
25 import org.w3c.dom.Element;
26 import org.w3c.dom.Node;
27 import org.w3c.dom.NodeList;
28
29 import java.io.InputStream;
30 import java.util.ArrayList;
31 import java.util.Locale;
32
33 import javax.xml.parsers.DocumentBuilder;
34 import javax.xml.parsers.DocumentBuilderFactory;
35
36 /**
37  * Created by Donisaurus on 4/25/2017.
38  */
39
40 public class PercakapanActivity extends AppCompatActivity
41 implements TextToSpeech.OnInitListener{
42     private RecyclerView rvPercakapan;
43     private VideoView vvPercakapan;
44     private TextToSpeech myTTS;
45     private int MY_DATA_CHECK_CODE = 0;
46

```

```

47     public void assignView(){
48         rvPercakapan =
49 (RecyclerView)findViewById(R.id.rvPercakapan);
50         vvPercakapan = (VideoView)findViewById(R.id.vvPercakapan);
51     }
52
53     @Override
54     protected void onCreate(Bundle savedInstanceState) {
55         super.onCreate(savedInstanceState);
56         setContentView(R.layout.activity_com_seharihari);
57
58         assignView();
59
60         //check for TTS data
61         Intent checkTTSIntent = new Intent();
62
63 checkTTSIntent.setAction(TextToSpeech.Engine.ACTION_CHECK_TTS_DATA
64 );
65         startActivityForResult(checkTTSIntent,
66 MY_DATA_CHECK_CODE);
67     }
68
69     //act on result of TTS data check
70     protected void onActivityResult(int requestCode, int
71 resultCode, Intent data) {
72
73         if (requestCode == MY_DATA_CHECK_CODE) {
74             if (resultCode ==
75 TextToSpeech.Engine.CHECK_VOICE_DATA_PASS) {
76                 //the user has the necessary data - create the TTS
77                 myTTS = new TextToSpeech(this, this);
78
79                 //Read XML from device
80                 PercakapanActivity.ReadXml readXml = new
81 PercakapanActivity.ReadXml(this, rvPercakapan, vvPercakapan,
82 myTTS);
83                 readXml.execute();
84             }
85             else {
86                 //no data - install it now
87                 Intent installTTSIntent = new Intent();
88
89 installTTSIntent.setAction(TextToSpeech.Engine.ACTION_INSTALL_TTS_
90 DATA);
91                 startActivity(installTTSIntent);
92             }
93         }
94     }
95
96     @Override
97     public void onInit(int initStatus) {
98         Locale loc= new Locale("ind","IDN");
99         //check for successful instantiation
100         if (initStatus == TextToSpeech.SUCCESS) {
101             if(myTTS.isLanguageAvailable(new
102 Locale("ind","IDN"))==TextToSpeech.LANG_AVAILABLE)
103                 myTTS.setLanguage(loc);1
104         }
105         else if (initStatus == TextToSpeech.ERROR) {
106             Toast.makeText(this, "Sorry! Text To Speech
107 failed...", Toast.LENGTH_LONG).show();
108         }
109     }
110
111     private class ReadXml extends AsyncTask<Void, Void, Void> {
112         Context context;
113         ProgressDialog progressDialog;
114         ArrayList<String> listPercakapan;

```

```

111     RecyclerView recyclerView;
112     VideoView vvPercakapan;
113     TextToSpeech myTTS;
114
115     public ReadXml(Context context, RecyclerView recyclerView,
116     VideoView vvPercakapan, TextToSpeech myTTS){
117         this.recyclerView = recyclerView;
118         this.context = context;
119         this.vvPercakapan = vvPercakapan;
120         this.myTTS = myTTS;
121         progressDialog = new ProgressDialog(context);
122         progressDialog.setMessage("Loading...");
123     }
124
125     @Override
126     protected void onPreExecute(){
127         progressDialog.show();
128         super.onPreExecute();
129     }
130
131     @Override
132     protected void onPostExecute(Void aVoid){
133         super.onPostExecute(aVoid);
134         progressDialog.dismiss();
135
136         PercakapanAdapter adapter = new
137     PercakapanAdapter(context, listPercakapan, vvPercakapan, myTTS);
138         recyclerView.setLayoutManager(new
139     LinearLayoutManager(context));
140         recyclerView.setAdapter(adapter);
141
142         // recyclerView.addItemTouchListener(new
143     RecyclerViewTouchListener(getApplicationContext(), recyclerView, new
144     ClickListener() {
145         // @Override
146         // public void onClick(View view, int position) {
147         //     Resources res = getResources();
148         //     int videoId =
149     res.getIdentifier(listPercakapan.get(position), "raw",
150     getPackageName());
151         //
152         // //Set Uri Video
153         // Uri videoPath =
154     Uri.parse("android.resource://" + getPackageName() + "/" +
155     videoId);
156         //
157         // vvPercakapan.setVideoURI(videoPath);
158         // vvPercakapan.start();
159         // }
160         // @Override
161         // public void onLongClick(View view, int position)
162         // {
163         //
164         // }
165         // }));
166     }
167
168     @Override
169     protected Void doInBackground(Void... params)
170     {
171         ProcessXml(GetData());
172         return null;
173     }
174
175     @Override
176     protected void onProgressUpdate(Void... text) {
177         //
178         // finalResult.setText(text[0]);
179     }
180
181     private void ProcessXml(Document data) {
182         if (data != null){
183             listPercakapan = new ArrayList<>();

```

```

202         Element root = data.getDocumentElement();
203         Node tabelkatatanya =
204 root.getChildNodes().item(1);
205         NodeList items = tabelkatatanya.getChildNodes();
206
207         for (int i = 0; i < items.getLength(); i++){
208             Node currentChild = items.item(i);
209             if
210 (currentChild.getNodeName().equalsIgnoreCase("value")){
211                 Log.d("nodes",
212 currentChild.getTextContent().substring(6,
213 currentChild.getTextContent().length()) + "");
214 listPercakapan.add(currentChild.getTextContent());
215             }
216         }
217     }else{
218         Log.d("Root", "gagal / data null");
219     }
220 }
221
222     public Document GetData(){
223         try {
224             InputStream inputStream =
225 getResources().openRawResource(R.raw.percakapan);
226             DocumentBuilderFactory builderFactory =
227 DocumentBuilderFactory.newInstance();
228             DocumentBuilder builder =
229 builderFactory.newDocumentBuilder();
230             Document xmlDoc = builder.parse(inputStream);
231             return xmlDoc;
232         } catch (Exception e){
233             e.printStackTrace();
234             return null;
235         }
236     }
237 }
238

```

Kode 0.4 Implementasi Kode Program Percakapan Sehari – Hari

Di bawah ini merupakan keterangan kode Program Percakapan Sehari-hari:

Nomer	Kode Program	Keterangan
78	<code>PercakapanActivity.ReadXml readXml = new PercakapanActivity.ReadXml(this, rvPercakapan, vvPercakapan, myTTS);</code>	Kode membaca dari Data XML dari <i>device</i>
215	<code>listPercakapan.add(currentChild.getTextContent());</code>	Kode <i>List</i> kata Percakapan Sehari-hari

5.4.5 Implementasi Kode Program SPOK

Implementasi Kode Program komunikasi *class* SpokActivity yang digunakan untuk menampilkan menu halaman pembelajaran pada aplikasi “**DekatTuli**” yang ditunjukkan pada Kode 5.5 sebagai berikut:

```

1 package com.example.donisaurus.DekatTuli.activity;
2
3 import android.app.ProgressDialog;
4 import android.content.Context;
5 import android.content.Intent;

```

```

5 import android.os.AsyncTask;
6 import android.os.Bundle;
7 import android.speech.tts.TextToSpeech;
8 import android.support.v7.app.AppCompatActivity;
9 import android.support.v7.widget.LinearLayoutManager;
10 import android.support.v7.widget.RecyclerView;
11 import android.support.v7.widget.SearchView;
12 import android.util.Log;
13 import android.view.View;
14 import android.widget.Button;
15 import android.widget.TextView;
16 import android.widget.Toast;
17
18 import com.example.donisaurus.DekatTuli.R;
19 import
20 com.example.donisaurus.DekatTuli.adapter.RecyclerTouchListener;
21 import com.example.donisaurus.DekatTuli.adapter.SpokAdapter;
22
23 import org.w3c.dom.Document;
24 import org.w3c.dom.Element;
25 import org.w3c.dom.Node;
26 import org.w3c.dom.NodeList;
27
28 import java.io.InputStream;
29 import java.lang.reflect.Array;
30 import java.util.ArrayList;
31 import java.util.Arrays;
32 import java.util.HashMap;
33 import java.util.List;
34 import java.util.Locale;
35
36 import javax.xml.parsers.DocumentBuilder;
37 import javax.xml.parsers.DocumentBuilderFactory;
38
39 /**
40  * Created by Donisaurus on 4/25/2017.
41  */
42
43 public class SpokActivity extends AppCompatActivity implements
44 TextToSpeech.OnInitListener{
45
46     private TextView tvSPOK;
47     private Button btnSpeak;
48     private SearchView searchSubject, searchPredicate,
49 searchObject, searchKeterangan;
50     private RecyclerView recyclerView1, recyclerView2,
51 recyclerView3, recyclerView4;
52     private TextToSpeech myTTS;
53     private int MY_DATA_CHECK_CODE = 0;
54
55     public void assignView(){
56         tvSPOK = (TextView)findViewById(R.id.tvSPOK);
57         btnSpeak = (Button)findViewById(R.id.btnSpeak);
58         searchSubject =
59 (SearchView)findViewById(R.id.searchSubject);
60         searchPredicate =
61 (SearchView)findViewById(R.id.searchPredicate);
62         searchObject =
63 (SearchView)findViewById(R.id.searchObject);
64         searchKeterangan =
65 (SearchView)findViewById(R.id.searchKeterangan);
66         recyclerView1 =
67 (RecyclerView)findViewById(R.id.recyclerView1);
68         recyclerView2 =
69 (RecyclerView)findViewById(R.id.recyclerView2);
70         recyclerView3 =
71 (RecyclerView)findViewById(R.id.recyclerView3);

```

```

72         recyclerView4 =
73 (RecyclerView) findViewById(R.id.recyclerView4);
74     }
75
76     @Override
77     protected void onCreate(Bundle savedInstanceState) {
78         super.onCreate(savedInstanceState);
79         setContentView(R.layout.activity_com_spok);
80
81         assignView();
82
83         //check for TTS data
84         Intent checkTTSIntent = new Intent();
85
86         checkTTSIntent.setAction(TextToSpeech.Engine.ACTION_CHECK_TTS_DATA
87 );
88         startActivityForResult(checkTTSIntent,
89 MY_DATA_CHECK_CODE);
90
91         btnSpeak.setOnClickListener(new View.OnClickListener() {
92             @Override
93             public void onClick(View v) {
94                 myTTS.speak(tvSPOK.getText().toString(),
95 TextToSpeech.QUEUE_FLUSH, null);
96             }
97         });
98
99         //act on result of TTS data check
100        protected void onActivityResult(int requestCode, int
101 resultCode, Intent data) {
102            if (requestCode == MY_DATA_CHECK_CODE) {
103                if (resultCode ==
104 TextToSpeech.Engine.CHECK_VOICE_DATA_PASS) {
105                    //the user has the necessary data - create the TTS
106                    myTTS = new TextToSpeech(this, this);
107
108                    //Read XML from device
109                    ReadXml readRss = new ReadXml(this, recyclerView1,
110 recyclerView2, recyclerView3, recyclerView4, tvSPOK);
111                    readRss.execute();
112                }
113                else {
114                    //no data - install it now
115                    Intent installTTSIntent = new Intent();
116
117                    installTTSIntent.setAction(TextToSpeech.Engine.ACTION_INSTALL_TTS_
118 DATA);
119                    startActivity(installTTSIntent);
120                }
121            }
122
123            @Override
124            public void onInit(int initStatus) {
125                Locale loc= new Locale("ind","IDN");
126                //check for successful instantiation
127                if (initStatus == TextToSpeech.SUCCESS) {
128                    if(myTTS.isLanguageAvailable(new
129 Locale("ind","IDN"))==TextToSpeech.LANG_AVAILABLE)
130                        myTTS.setLanguage(loc);
131                }
132                else if (initStatus == TextToSpeech.ERROR) {
133                    Toast.makeText(this, "Sorry! Text To Speech
134 failed...", Toast.LENGTH_LONG).show();
135                }
136            }
137        }

```

```

135
136     private class ReadXml extends AsyncTask<Void, Void, Void> {
137         Context context;
138         ProgressDialog progressDialog;
139         List<String> subyek;
140         List<String> predikat;
9         List<String> obyek;
170         List<String> keterangan;
171         RecyclerView recyclerView1, recyclerView2, recyclerView3,
recyclerView4;
172         TextView tvSPOK;
173         String[] spok = {"", "", "", ""};
174
175         public ReadXml(Context context, RecyclerView
176 recyclerView1, RecyclerView recyclerView2, RecyclerView
177 recyclerView3, RecyclerView recyclerView4, TextView tvSPOK){
178             this.recyclerView1 = recyclerView1;
179             this.recyclerView2 = recyclerView2;
180             this.recyclerView3 = recyclerView3;
181             this.recyclerView4 = recyclerView4;
182             this.tvSPOK = tvSPOK;
183
184             this.context = context;
185             progressDialog = new ProgressDialog(context);
186             progressDialog.setMessage("Loading...");
187         }
188
189         @Override
190         protected void onPreExecute(){
191             progressDialog.show();
192             super.onPreExecute();
193         }
194
195         @Override
196         protected void onPostExecute(Void aVoid){
197             super.onPostExecute(aVoid);
198             progressDialog.dismiss();
199
200             //Subyek adapter
201             SpokAdapter subyekAdapter = new SpokAdapter(context,
subyek);
202             recyclerView1.setLayoutManager(new
LinearLayoutManager(context));
203             recyclerView1.setAdapter(subyekAdapter);
204             recyclerView1.setOnItemClickListener(new
RecyclerViewTouchListener(getApplicationContext(), recyclerView1, new
ClickListener() {
205                 @Override
206                 public void onClick(View view, int position) {
207                     String current = subyek.get(position);
208                     spok[0] = current;
209                     String content =
210                     Arrays.toString(spok).substring(1, Arrays.toString(spok).length()-
211                     1).replace(',', ' ');
212                     tvSPOK.setText(content);
213                 }
214             }
215
216             @Override
217             public void onLongClick(View view, int position) {
218             }
219         }
220     });
221
222     //Predikat adapter
223     SpokAdapter predikatAdapter = new SpokAdapter(context,
predikat);
224     recyclerView2.setLayoutManager(new
LinearLayoutManager(context));
225

```



```

226         recyclerView2.setAdapter(predikatAdapter);
227         recyclerView2.setOnItemClickListener(new
228 RecyclerViewTouchListener(getApplicationContext(), recyclerView2, new
229 ClickListener() {
230             @Override
231             public void onClick(View view, int position) {
232                 String current = predikat.get(position);
233                 spok[1] = current;
234                 String content =
235 Arrays.toString(spok).substring(1, Arrays.toString(spok).length()-
236 1).replace(',', ' ');
237                 tvSPOK.setText(content);
238             }
239             @Override
240             public void onLongClick(View view, int position) {
241             }
242         });
243     });
244     //Obyek adapter
245     SpokAdapter obyekAdapter = new SpokAdapter(context,
246 obyek);
247     recyclerView3.setLayoutManager(new
248 LinearLayoutManager(context));
249     recyclerView3.setAdapter(obyekAdapter);
250     recyclerView3.setOnItemClickListener(new
251 RecyclerViewTouchListener(getApplicationContext(), recyclerView3, new
252 ClickListener() {
253         @Override
254         public void onClick(View view, int position) {
255             String current = obyek.get(position);
256             spok[2] = current;
257             String content =
258 Arrays.toString(spok).substring(1, Arrays.toString(spok).length()-
259 1).replace(',', ' ');
260             tvSPOK.setText(content);
261         }
262         @Override
263         public void onLongClick(View view, int position) {
264         }
265     });
266     });
267     //Keterangan adapter
268     SpokAdapter keteranganAdapter = new
269 SpokAdapter(context, keterangan);
270     recyclerView4.setLayoutManager(new
271 LinearLayoutManager(context));
272     recyclerView4.setAdapter(keteranganAdapter);
273     recyclerView4.setOnItemClickListener(new
274 RecyclerViewTouchListener(getApplicationContext(), recyclerView4, new
275 ClickListener() {
276         @Override
277         public void onClick(View view, int position) {
278             String current = keterangan.get(position);
279             spok[3] = current;
280             String content =
281 Arrays.toString(spok).substring(1, Arrays.toString(spok).length()-
282 1).replace(',', ' ');
283             tvSPOK.setText(content);
284         }
285         @Override
286         public void onLongClick(View view, int position) {
287         }
288     });

```

```

289         });
290     }
291
292     @Override
293     protected Void doInBackground(Void... params)
294     {
295         ProcessXml(GetData());
296         return null;
297     }
298
299     @Override
300     protected void onProgressUpdate(Void... text) {
301         //         finalResult.setText(text[0]);
302     }
303
304     private void ProcessXml(Document data) {
305         if (data != null){
306             subyek = new ArrayList<>();
307             predikat = new ArrayList<>();
308             obyek = new ArrayList<>();
309             keterangan = new ArrayList<>();
310
311             Element root = data.getDocumentElement();
312             NodeList tabelspok = root.getChildNodes();
313
314             for (int i = 0; i < tabelspok.getLength(); i++){
315                 Node item = tabelspok.item(i);
316                 NodeList items = item.getChildNodes();
317                 for (int j = 0; j < items.getLength(); j++){
318                     Node currentChild = items.item(j);
319                     if
320                     (currentChild.getNodeName().equalsIgnoreCase("subyek") &&
321                     currentChild.getTextContent() != ""){
322                         Log.d("nodes",
323                         currentChild.getTextContent());
324                         subyek.add(currentChild.getTextContent());
325                     }else if
326                     (currentChild.getNodeName().equalsIgnoreCase("predikat") &&
327                     currentChild.getTextContent() != ""){
328                         Log.d("nodes",
329                         currentChild.getTextContent());
330                         predikat.add(currentChild.getTextContent());
331                     }else if
332                     (currentChild.getNodeName().equalsIgnoreCase("obyek") &&
333                     currentChild.getTextContent() != ""){
334                         Log.d("nodes",
335                         currentChild.getTextContent());
336                         obyek.add(currentChild.getTextContent());
337                     }else if
338                     (currentChild.getNodeName().equalsIgnoreCase("keterangan") &&
339                     currentChild.getTextContent() != ""){
340                         Log.d("nodes",
341                         currentChild.getTextContent());
342                         keterangan.add(currentChild.getTextContent());
343                     }
344                 }
345             }
346
347             Log.d("panjang", tabelspok.getLength() + "");
348         }else{
349             Log.d("Root", "gagal / data null");
350         }
351     }

```

```

352     public Document GetData(){
353         try {
354             InputStream inputStream =
339 getResources().openRawResource(R.raw.spok);
339         DocumentBuilderFactory builderFactory =
339 DocumentBuilderFactory.newInstance();
339         DocumentBuilder builder =
339 builderFactory.newDocumentBuilder();
339         Document xmlDoc = builder.parse(inputStream);
339         return xmlDoc;
339     } catch (Exception e){
339         e.printStackTrace();
339         return null;
339     }
339 }
339
339     public String getValue(String tag, Element element) {
339         NodeList nodeList =
339 element.getElementsByTagName(tag).item(0).getChildNodes();
339         Node node = nodeList.item(0);
339         return node.getNodeValue();
339     }
339 }
339 }
339
339
339
339

```

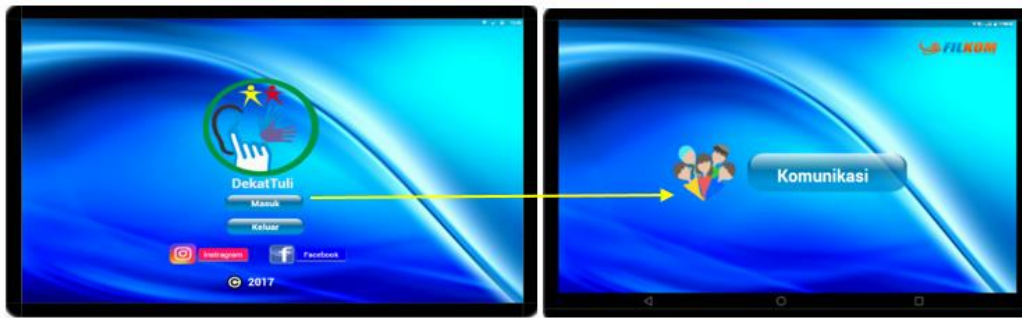
Kode 0.5 Implementasi Kode Program SPOK

Di bawah ini merupakan keterangan kode Program SPOK:

Nomer	Kode Program	Keterangan
108	ReadXml readRss = new ReadXml(this, recyclerView1, recyclerView2, recyclerView3, recyclerView4, tvSPOK);readRss.execute();	Kode membaca dari Data XML dari <i>device</i>
199	SpokAdapter subyekAdapter = new SpokAdapter(context, subyek);	Kode Subjek Adapter
223	SpokAdapter predikatAdapter = new SpokAdapter(context, predikat);	Kode Predikat Adapter
246	SpokAdapter obyekAdapter = new SpokAdapter(context, obyek);	Kode Objek Adapter
269	SpokAdapter keteranganAdapter = new SpokAdapter(context, keterangan);	Kode Keterangan Adapter

5.5 Implementasi Antarmuka

Implementasi antarmuka menampilkan hasil implementasi dari perancangan antarmuka aplikasi perangkat bergerak dalam Gambar 4.15 pada bab 4. Penjelasan implementasi antarmuka aplikasi perangkat bergerak yang dibangun dapat dilihat dalam Gambar 5.2 sampai Gambar 5.6.



Gambar 0.2 Implementasi Antarmuka Aplikasi Menu Utama



GGambar 0.3 Implementasi Antarmuka pilihan Kata Tanya



Gambar 0.4 Implementasi Antarmuka pilihan komunikasi menu



Gambar 0.5 Implementasi Antarmuka komunikasi percakapan Sehari- hari



Gambar 0.6 Implementasi Antarmuka komunikasi SPOK