

Aplikasi Android Kamus Idiom Bahasa Manado - Malayu

Andrew Wuwungan, Yaulie D. Y. Rindengan , Xaverius B. N. Najoan
Jurusan Teknik Elektro, Universitas Sam Ratulangi Manado, Jl. Kampus Bahu, 95115, Indonesia
e-mail : awuwungan16@gmail.com, rindengan@unsrat.ac.id, xnajoan@unsrat.ac.id
Diterima: tgl; direvisi: tgl; disetujui: tgl

Abstrak — Kemajuan teknologi pada bidang Elektronika berkembang sangat cepat dan sangat berpengaruh dalam kehidupan. Salah satunya adalah Bahasa, yang sangat dibutuhkan oleh wisatawan maupun anak muda zaman sekarang untuk berkomunikasi. Pengetahuan tentang Bahasa sangatlah penting karena dalam sebuah pembicaraan atau percakapan memerlukan Bahasa. Maka dari itu dibuat skripsi dengan judul “ Aplikasi Android Kamus Idiom Bahasa Manado-Melayu”. Aplikasi ini dibangun menggunakan Android Studio dan databasenya menggunakan SQLite.

Metode yang digunakan dalam membangun Aplikasi ini adalah *Extreme Programming (XP)*. Metode ini terbagi menjadi 4 tahapan yaitu, *Plenning (Perencanaan)*, *Design (Perancangan)*, *Coding (Pengkodean)*, *Testing (Pengujian)*.

Dari hasil pembahasan maka dapat disimpulkan bahwa dengan aplikasi ini dapat mempermudah para wisatawan maupun anak muda zaman sekarang untuk mengenal Bahasa Indiom Manado-Melayu. Dikarenakan pada era moderen saat ini Bahasa sangatlah dibutuhkan untuk percakapan sehari-hari.

Kata kunci — Bahasa Manado-Melayu, Extreme Programming, SQLite

Abstract — Technological advances in the field of Electronics are growing very fast and are very influential in life. One of them is language, which is really needed by tourists and young people today to communicate. Knowledge of language is very important because in a conversation or conversation requires language. Therefore, a thesis is made with the title "Android Application of the Manado-Malay Language Idiom Dictionary". This application is built using Android Studio and the database uses SQLite.

The method used in building this application is *Extreme Programming (XP)*. This method is divided into 4 stages, namely, *Plenning (Planning)*, *Design (Design)*, *Coding (Encoding)*, *Testing (Testing)*.

From the results of the discussion, it can be concluded that this application can make it easier for tourists and young people today to get to know the Manado-Malay Indiom Language. Because in the modern era, language is needed for daily conversation.

Keywords — Manado-Malay Language, Extreme Programming, SQLite

I. PENDAHULUAN

Kemajuan teknologi pada bidang elektronika berkembang sangat cepat dan sangat berpengaruh dalam kehidupan. Teknologi sendiri memiliki pengertian yaitu menyediakan barang-barang yang diperlukan bagi keberlangsungan hidup manusia. Bahasa merupakan sarana untuk berkomunikasi. Pengetahuan tentang bahasa sangatlah penting karena dalam sebuah percakapan atau pembicaraan memerlukan bahasa. Bahasa sangat dibutuhkan dalam kehidupan sehari-hari,

karena dalam kehidupan sosial bahasa sangat dibutuhkan dan sangat diperlukan untuk saling berkomunikasi.

Aplikasi Kamus Idiom Bahasa Manado-Melayu ini dibangun menggunakan Android Studio, dan aplikasi ini dibuat agar masyarakat, maupun wisatawan dapat menggunakannya untuk dapat berbahasa Manado-Melayu.

II. LANDASAN TEORI

A. Pengertian Idiom

Idiom merupakan kata lain dari ungkapan. Idiom adalah gabungan kata yang bermakna khusus. Makna dari idiom bukanlah berasal dari kata-kata

pembentuknya, akan tetapi terbentuk setelah kata-kata tersebut digabungkan. Idiom sering digunakan sebagai kiasan dalam penyampaian makna.

B. Kamus Elektronik

Kamus Elektronik adalah kamus digital yang bentuknya berupa software atau aplikasi. “E-Kamus” bisa berbentuk software yang perlu diinstal dikomputer (berbasis komputer), atau diletakkan di laman (halaman) *website*, atau berupa aplikasi yang kini dapat dioperasikan melalui perangkat *mobile* seperti ponsel, tablet, dan sebagainya.

C. SQLite

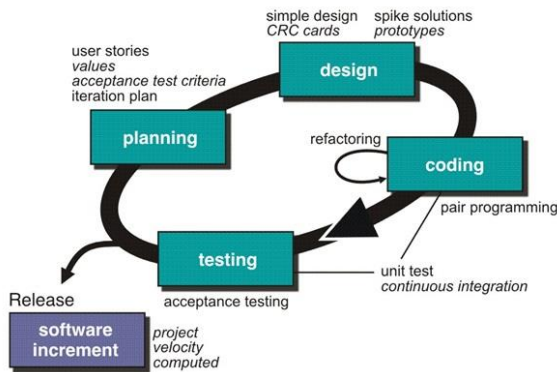
SQLite adalah perpustakaan perangkat lunak yang menerapkan engine database SQL secara mandiri, tanpa memerlukan server, tanpa perlu melakukan konfigurasi, dan bersifat transaksional. SQLite adalah engine database SQL yang paling banyak digunakan di dunia. SQLite merupakan proyek yang bersifat public domain yang dikerjakan oleh D. Richard Hipp.

D. Java

Java adalah merupakan bahasa pemrograman yang disusun oleh James Gosling yang dibantu oleh rekan – rekannya di suatu perusahaan perangkat lunak yang bernama *Sun Microsystems*, pada tahun 1991. Menurut definisi dari *Sun Microsystems*, di dalam buku M. Shalahuddin dan Rosa A.S., Java adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri ataupun pada lingkungan jaringan.

III. METODE

Metode yang digunakan dalam pembuatan aplikasi yaitu metode XP (*Extreme Programming*) dimana metode ini adalah metode pengembangan aplikasi yang menitik beratkan pada aktivitas implementasi program.



Gambar 1. Metode *Extreme Programming*

IV. HASIL DAN PEMBAHASAN

1) *Planning*

Pada tahap ini yang dilakukan adalah mencari dan melihat sumber berupa jurnal atau skripsi, buku dokumentasi, internet dan pustaka lainnya tentang bahasa Manado-Melayu. Untuk mengimplementasikan pada aplikasi yang ingin dibuat.

2) *Design*

A. Storyboard

Pada tahap pertama ini adalah membuat design atau membuat storyboard aplikasi ini :



Gambar 2. Storyboard Tampilan Utama



Gambar 3. Tampilan Mencari Kata

B. Pemodelan Aplikasi Dengan Android Studio

Dan pada tahap ke dua adalah membuat tampilan dengan menggunakan android studio.



Gambar 4. Tampilan Awal Aplikasi

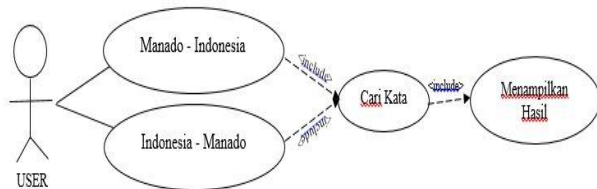


Gambar 5. Tampilan Memilih Bahasa

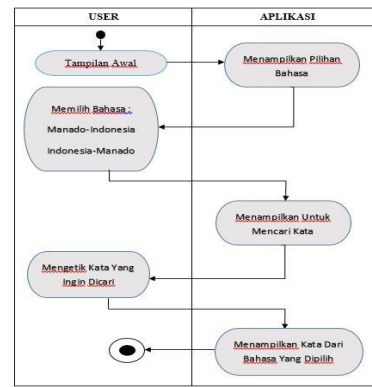
A. Pemodelan Sistem

Use Case Diagram adalah dialog antara sistem dan user, dalam bertukar pesan dan tindakan yang dilakukan oleh sistem.

Activity diagram merupakan alur berjalannya aplikasi sesuai urutan aktivitas.



Gambar 6. Use Case Diagram



Gambar 7. Activity Diagram

3) Coding

```

1 package com.uya.drawerlayout;
2
3 import android.support.annotation.NonNull;
4 import android.support.design.widget.NavigationView;
5 import android.support.v4.app.Fragment;
6 import android.support.v4.app.FragmentManager;
7 import android.support.v4.widget.DrawerLayout;
8 import android.support.v7.app.ActionBarDrawerToggle;
9 import android.support.v7.app.AppCompatActivity;
10 import android.os.Bundle;
11 import android.view.MenuItem;
12
13 public class MainActivity extends AppCompatActivity {
14
15     private DrawerLayout mDrawerLayout;
16     private ActionBarDrawerToggle mToggle;
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_main);
22         mDrawerLayout = (DrawerLayout) findViewById(R.id.drawer);
23         mToggle = new ActionBarDrawerToggle(this, mDrawerLayout, R.string.open, R.string.close);
24         mDrawerLayout.addDrawerListener(mToggle);
25         mToggle.syncState();
26         NavigationView mDrawer = (NavigationView) findViewById(R.id.navMenu);
27         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
28         setupDrawerContent(mDrawer);
29         switchToFragment(dashboard.class);
30     }
31
32     public void selectItemDrawer(MenuItem menuItem){
33         switch (menuItem.getItemId()){
34             case R.id.db:
35                 switchToFragment(dashboard.class);
36                 break;
37             case R.id.tambahKata:
38                 switchToFragment(TambahKata.class);
39                 break;
40             case R.id.daftarKata:
41                 switchToFragment(DaftarKata.class);
42                 break;
43             case R.id.valuar:
44                 finish();
45                 break;
46         }
47
48         menuItem.setChecked(true);
49         setTitle(menuItem.getTitle());
50         mDrawerLayout.closeDrawers();
51     }
52
53     private void switchToFragment(Class fragmentClass){
54         Fragment myFragment = null;
55         try {
56             myFragment = (Fragment) fragmentClass.newInstance();
57         } catch (Exception e){
58             e.printStackTrace();
59         }
60         FragmentManager fragmentManager = getSupportFragmentManager();
61         fragmentManager.beginTransaction().replace(R.id.flContent, myFragment).commit();
62     }
63
64     private void setupDrawerContent(NavigationView navigationView){
65         navigationView.setNavigationItemSelectedListener(new NavigationView.OnNavigationItemSelectedListener() {
66             @Override
67             public boolean onNavigationItemSelected(@NonNull MenuItem item) {
68                 selectItemDrawer(item);
69                 return true;
70             }
71         });
72     }
73
74     @Override
75     public boolean onOptionsItemSelected(MenuItem item) {
76         if (mToggle.onOptionsItemSelected(item)){
77             return true;
78         }
79         return super.onOptionsItemSelected(item);
80     }
81
82 }
83
84
    
```

Gambar 8. MainActivity.java

```
1 package com.uys.drawerlayout;
2
3 import android.app.AlertDialog;
4 import android.content.Context;
5 import android.content.DialogInterface;
6 import android.database.Cursor;
7 import android.database.sqlite.SQLiteDatabase;
8 import android.net.Uri;
9 import android.os.Bundle;
10 import android.support.annotation.Nullable;
11 import android.support.v4.app.Fragment;
12 import android.util.Log;
13 import android.view.ContextMenu;
14 import android.view.LayoutInflater;
15 import android.view.Menu;
16 import android.view.MenuItem;
17 import android.view.View;
18 import android.view.ViewGroup;
19 import android.widget.AdapterView;
20 import android.widget.AdapterView.OnItemClickListener;
21 import android.widget.SimpleCursorAdapter;
22 import android.widget.TextView;
23
24
25 /**
26  * A simple {@link Fragment} subclass.
27  * Activities that contain this Fragment must implement the
28  * {@link DaftarKata.OnFragmentInteractionListener} interface
29  * to handle interaction events.
30  * Use the {@link DaftarKata.newInstance} factory method to
31  * create an instance of this fragment.
32  */
33 public class DaftarKata extends Fragment {
34
35     private Cursor kamusCursor = null;
36     private CustomCursorAdapter adapter;
37     private DatabaseHelper dbHelper;
38     private SQLiteDatabase db = null;
39
40     private ListView listView = null;
41     private DatabaseHelper database = null;
42     private static final int DELETE_ID = Menu.FIRST + 1;
43
44     public void onCreate(Bundle savedInstanceState) {
45         super.onCreate(savedInstanceState);
46         dbHelper = new DatabaseHelper(getActivity());
47         listView = (ListView) getActivity().findViewById(R.id.listView);
48         adapter = new CustomCursorAdapter(getActivity(), R.layout.row, kamusCursor, from, to);
49         registerForContextMenu(listView);
50     }
51
52     private void setUpListView() {
53         try {
54             db = dbHelper.getWritableDatabase();
55             kamusCursor = db.rawQuery("select * from 'kamus'", null);
56             //format array to specify the fields want to display in the list
57             String[] from = new String[] {
58                 "id", "id_kali",
59             };
60             //array of the fields we want to bind those fields to (in this case just the textView 'id' and 'id_kali' from our new row.xml layout above)
61             int[] to = new int[] { R.id.id_indonesia, R.id.kali};
62             adapter = new CustomCursorAdapter(getActivity(), R.layout.row, kamusCursor, from, to);
63             listView.setAdapter(adapter);
64             listView.setOnItemClickListener(adapter);
65         } catch (Exception e) {
66             Log.e("DaftarKata", "Error");
67         }
68     }
69
70     protected class CustomCursorAdapter extends SimpleCursorAdapter {
71         private int layout;
72         private LayoutInflater inflater;
73         private Context context;
74
75         public CustomCursorAdapter(Context context, int layout, Cursor c, String[] from, int[] to) {
76             super(context, layout, c, from, to);
77             this.layout = layout;
78             this.context = context;
79             inflater = LayoutInflater.from(context);
80         }
81
82         @Override
83         public View newView(Context context, Cursor cursor, ViewGroup parent) {
84             return super.newView(context, cursor, parent);
85             Log.d("DaftarKata", "newView");
86             View v = inflater.inflate(R.layout.row, parent, false);
87             return v;
88         }
89
90         @Override
91         public void bindView(View view, Context context, Cursor cursor) {
92             //super.bindView(view, context, cursor);
93             //1 is the column where you're getting your data from
94             String indonesia = cursor.getString(1);
95             String kali = cursor.getString(2);
96
97             //Next set the name of the entry.
98             TextView ind_text = (TextView) view.findViewById(R.id.indonesia);
99             TextView kali_text = (TextView) view.findViewById(R.id.kali);
100             TextView id_text = (TextView) view.findViewById(R.id.id_indonesia);
101             ind_text.setText(indonesia);
102             kali_text.setText(kali);
103         }
104     }
105
106     @Override
107     public void onCreateContextMenu(Context menu, View v, ContextMenu.ContextMenuInfo menuInfo) {
108         super.onCreateContextMenu(menu, v, menuInfo);
109         menu.add(Menu.NONE, DELETE_ID, Menu.NONE, "hapus").setIcon(R.drawable.ic_launcher_background).setAlphaabeticShortcut("h");
110     }
111
112     @Override
113     public boolean onContextItemSelected(MenuItem item) {
114         switch (item.getItemId()) {
115             case DELETE_ID:
116                 AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo) item.getMenuInfo();
117                 delete(info.id);
118                 return (true);
119         }
120         return (super.onContextItemSelected(item));
121     }
122
123     private void delete(final long rowId) {
124         if (rowId > 0) {
125             new AlertDialog.Builder(getActivity())
126                 .setTitle(R.string.delete_title)
127                 .setPositiveButton(R.string.ok, new DialogInterface.OnClickListener() {
128                     @Override
129                     public void onClick(DialogInterface dialogInterface, int i) {
130                         processDelete(rowId);
131                     }
132                 })
133                 .setNegativeButton(R.string.cancel, new DialogInterface.OnClickListener() {
134                     @Override
135                     public void onClick(DialogInterface dialogInterface, int i) {
136                         //ignore, just dismiss
137                     }
138                 })
139                 .show();
140         }
141     }
142
143     private void processDelete(long rowId) {
144         String[] args = { String.valueOf(rowId)};
145         db.delete("kamus", "_id=?", args);
146         kamusCursor.requery();
147     }
148
149     // TODO: Rename parameter arguments, choose names that match
150     // the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
151     private static final String ARG_PARAM1 = "param1";
152     private static final String ARG_PARAM2 = "param2";
153
154     // TODO: Rename and change types of parameters
155     private String mParam1;
156     private String mParam2;
157
158     private OnFragmentInteractionListener mListener;
159
160     public DaftarKata() {
161         // Required empty public constructor
162     }
163
164     // TODO: Rename and change types and number of parameters
165     public static DaftarKata newInstance(String param1, String param2) {
166         DaftarKata fragment = new DaftarKata();
167         Bundle args = new Bundle();
168         args.putString(ARG_PARAM1, param1);
169         args.putString(ARG_PARAM2, param2);
170         fragment.setArguments(args);
171         return fragment;
172     }
173
174     @Override
175     public void onCreate(Bundle savedInstanceState) {
176         super.onCreate(savedInstanceState);
177         if (getArguments() != null) {
178             mParam1 = getArguments().getString(ARG_PARAM1);
179             mParam2 = getArguments().getString(ARG_PARAM2);
180         }
181     }
182
183     @Override
184     public void onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
185         // Inflate the layout for this fragment
186         return inflater.inflate(R.layout.fragment_daftar_kata, container, false);
187     }
188
189     // TODO: Rename method, update argument and hook method into UI event
190     public void onAttach(Context context) {
191         if (mListener != null) {
192             mListener.onFragmentInteraction(uri);
193         }
194     }
195
196     @Override
197     public void onDetach() {
198         super.onDetach();
199     }
200
201     @Override
202     public void onCreate(Bundle savedInstanceState) {
203         super.onCreate(savedInstanceState);
204     }
205
206     /**
207      * This interface must be implemented by activities that contain this
208      * fragment to allow an interaction in this fragment to be communicated
209      * to the activity and potentially other fragments contained in that
210      * activity.
211      * See the Android Training Lesson <a href="
212      * https://developer.android.com/courses/essential/android-basics/communicating.html"
213      * >>
214      * Communicating with Other Fragments</a> for more information.
215      */
216     public interface OnFragmentInteractionListener {
217         // TODO: Update argument type and name
218         void onFragmentInteraction(Uri uri);
219     }
220 }
221
222 package com.uys.drawerlayout;
223
224 import android.content.ContentValues;
225 import android.content.Context;
226 import android.database.Cursor;
227 import android.database.DatabaseErrorHandler;
228 import android.database.sqlite.SQLiteDatabase;
229 import android.database.sqlite.SQLiteOpenHelper;
230 import android.util.Log;
231 import android.widget.Toast;
232
233 import java.io.BufferedReader;
234 import java.io.FileNotFoundException;
235 import java.io.FileReader;
236 import java.io.IOException;
237 import java.io.InputStream;
238 import java.io.InputStreamReader;
239 import java.nio.charset.Charset;
240
241 /**
242  * Created by root on 14/03/18.
243  */
244
245 public class DatabaseHelper extends SQLiteOpenHelper {
246
247     private static final String DATABASE_NAME = "dbkamus";
248     private static final String INDONESIA = "indonesia";
249     private static final String KALI = "kali";
250     private Context mContext;
251     private Context mBaseContext;
252
253     public DatabaseHelper(Context context, Context baseContext) {
254         super(context, DATABASE_NAME, null, 1);
255         mContext = context;
256         mBaseContext = baseContext;
257     }
258 }
```

Gambar 9. DaftarKata.java

```

39 public void createTable(SQLiteDatabase db) {
40     db.execSQL("CREATE TABLE IF NOT EXISTS kamaus");
41     db.execSQL("CREATE TABLE IF NOT EXISTS kamaus (_id INTEGER PRIMARY KEY AUTOINCREMENT, * * Indonesia TEXT, Kalll TEXT);");
42
43
44
45
46 public void urutData(SQLiteDatabase db) {
47     Cursor kamausCursor;
48     kamausCursor = db.rawQuery("SELECT * FROM kamaus ORDER BY Indonesia ASC", null);
49     String[][] array = new String[kamausCursor.getCount()][3]; // Dynamic string array
50     int i = 0;
51     if (kamausCursor.moveToFirst()) {
52         for (; kamausCursor.isAfterLast(); kamausCursor.moveToNext()) {
53             array[i][0] = kamausCursor.getString(0);
54             array[i][1] = kamausCursor.getString(1);
55             array[i][2] = kamausCursor.getString(2);
56             i++;
57         }
58     } else {
59         Toast.makeText(CurrentBaseContext, "Data Tidak Ada",
60             Toast.LENGTH_SHORT).show();
61     }
62
63     kamausCursor = db.rawQuery("SELECT * FROM kamaus", null);
64     ContentValues values = new ContentValues();
65     int i = 0;
66     if (kamausCursor.moveToFirst()) {
67         for (; kamausCursor.isAfterLast(); kamausCursor.moveToNext()) {
68             values.put("Indonesia", array[i][0]);
69             values.put("Kalll", array[i][1]);
70             String[] args = {String.valueOf(kamausCursor.getString(0))};
71             db.update("kamaus", values, "_id=?", args);
72             i++;
73         }
74     }
75
76
77     kamausCursor.requery();
78     Toast.makeText(CurrentBaseContext, "Urut Data Berhasil", Toast.LENGTH_SHORT).show();
79
80
81
82
83 public void generateData(SQLiteDatabase db) {
84     String csvFile = "home/subhandp/KAMUS_ANDROID/dataKamus.csv";
85
86     BufferedReader br = null;
87     String line = "";
88     String cvsSplitBy = ",";
89
90     Log.i("CSVTEST", "MULAI");
91     ContentValues cv = new ContentValues();
92     try {
93         //br = new BufferedReader(new FileReader(csvFile));
94         //InputStream is = CurrentContext.getResources().openRawResource(R.raw.dataKamus);
95         br = new BufferedReader(
96             new InputStreamReader(is, Charset.forName("UTF-8")));
97
98         while (line = br.readLine() != null) {
99             // Use comma as separator
100             String[] dataKamus = line.split(cvsSplitBy);
101
102             cv.put("INDONESIA", dataKamus[0].toLowerCase());
103             cv.put("KALLL", dataKamus[1].toLowerCase());
104             db.insert("kamaus", null, cv);
105         }
106     } catch (FileNotFoundException e) {
107         e.printStackTrace();
108         Log.e("CSVTEST", "ERROR OPEN");
109     } catch (IOException e) {
110         e.printStackTrace();
111         Log.e("CSVTEST", "ERROR OPEN");
112     } finally {
113         if (br != null) {
114             try {
115                 br.close();
116             } catch (IOException e) {
117                 e.printStackTrace();
118                 Log.e("CSVTEST", "ERROR OPEN");
119             }
120         }
121     }
122
123
124
125
126
127
128
129
130 @Override
131 public void onCreate(SQLiteDatabase sqLiteDatabase) {
132     createTable(sqLiteDatabase);
133     generateData(sqLiteDatabase);
134     urutData(sqLiteDatabase);
135
136
137 @Override
138 public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
139     createTable(sqLiteDatabase);
140     generateData(sqLiteDatabase);
141     urutData(sqLiteDatabase);
142
143
144

```

```

39 private ImageButton btnSpeechToText;
40 private ImageButton btnTextToSpeech;
41 private ImageButton btnImageClear;
42 private TextToSpeech textToSpeech;
43 private final int REQ_CODE_SPEECH_INPUT = 100;
44 private DatabaseHelper dataKamus = null;
45
46 public static int binarySearch(String[][] a, String x) {
47     int low = 0;
48     int high = a.length - 1;
49     int mid;
50
51     while (low <= high) {
52         mid = (low + high) / 2;
53         String kataDb = a[mid][1].toLowerCase();
54         String cari = x.toLowerCase();
55         if (kataDb.compareTo(cari) < 0) {
56             low = mid + 1;
57         } else if (kataDb.compareTo(cari) > 0) {
58             high = mid - 1;
59         } else {
60             return mid;
61         }
62     }
63     return -1;
64
65
66 public void getTerjemahan(View view) {
67     String bhkalll = "";
68     String indonesiaInput = EditTextIndonesia.getText().toString();
69
70
71     kamausCursor = db.rawQuery("SELECT * FROM kamaus", null);
72     String[][] array = new String[kamausCursor.getCount()][3]; // Dynamic string array
73     int i = 0;
74     if (kamausCursor.moveToFirst()) {
75         for (; kamausCursor.isAfterLast(); kamausCursor.moveToNext()) {
76             array[i][0] = kamausCursor.getString(0);
77             array[i][1] = kamausCursor.getString(1);
78             array[i][2] = kamausCursor.getString(2);
79             i++;
80         }
81     }
82
83     indonesiaInput = indonesiaInput.toLowerCase();
84     int low = binarySearch(array, indonesiaInput);
85     if (low != -1) {
86         bhkalll = array[low][2];
87         txtKalll.setVisibility(View.VISIBLE);
88         btnTextToSpeech.setVisibility(View.VISIBLE);
89     } else {
90         txtKalll.setVisibility(View.INVISIBLE);
91         btnTextToSpeech.setVisibility(View.INVISIBLE);
92
93         AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
94         builder.setMessage("Data Tidak Ada.");
95         builder.setCancelable(true);
96
97         builder.setPositiveButton(
98             "OK",
99             new DialogInterface.OnClickListener() {
100                 public void onClick(DialogInterface dialog, int id) {
101                     dialog.cancel();
102                 }
103             });
104
105         AlertDialog alert11 = builder.create();
106         alert11.show();
107     }
108
109     Toast.makeText(getActivity().getBaseContext(), "Data Tidak Ada",
110         Toast.LENGTH_SHORT).show();
111
112
113     txtKalll.setText(bhkalll);
114
115
116
117
118
119
120
121
122 // Untuk menampilkan Google speech input dialog
123 public void tanyaInputSuara() {
124     Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
125     //Intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, new Locale("id", "ID"));
126     Intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
127         RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
128     Intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, "id-ID");
129     Intent.putExtra(RecognizerIntent.EXTRA_PROMPT,
130         "Hai Bicara sekarang");
131
132     try {
133         startActivityForResult(intent, REQ_CODE_SPEECH_INPUT);
134     } catch (ActivityNotFoundException e) {
135     }
136
137
138 @Override
139 public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {
140     super.onViewCreated(view, savedInstanceState);
141     dataKamus = new DatabaseHelper(getActivity(), getActivity().getBaseContext());
142     db = dataKamus.getSQLiteDatabase();
143     EditTextIndonesia = (EditText) getActivity().findViewById(R.id.EditTextIndonesia);
144     txtKalll = (TextView) getActivity().findViewById(R.id.txtKalll);
145     btnTerjemahan = (Button) view.findViewById(R.id.btnTerjemahan);
146     btnSpeechToText = (ImageButton) view.findViewById(R.id.imageBtnSpeechText);
147     btnTextToSpeech = (ImageButton) view.findViewById(R.id.imageBtnTextSpeech);
148     btnImageClear = (ImageButton) view.findViewById(R.id.imageClear);
149
150     txtKalll.setVisibility(View.INVISIBLE);
151     btnTextToSpeech.setVisibility(View.INVISIBLE);
152     btnImageClear.setVisibility(View.INVISIBLE);
153
154     EditTextIndonesia.addTextChangedListener(new dashboard.MyTextWatcher(EditTextIndonesia));
155
156     btnTerjemahan.setOnClickListener(new View.OnClickListener() {
157         @Override
158         public void onClick(View v) {
159             getTerjemahan(v);
160         }
161     });
162
163     btnSpeechToText.setOnClickListener(new View.OnClickListener() {
164         @Override
165         public void onClick(View view) {
166             tanyaInputSuara();
167         }
168     });
169
170     textToSpeech = new TextToSpeech(getActivity(), new TextToSpeech.OnInitListener() {
171         @Override
172         public void onInit(int status) {
173             //method untuk mendeteksi suara dari text
174             if (status == TextToSpeech.ERROR) {
175                 textToSpeech.setLanguage(new Locale("id", "ID"));
176             }
177         }
178     });
179
180     btnTextToSpeech.setOnClickListener(new View.OnClickListener() {
181         @Override
182         public void onClick(View view) {
183             String text = txtKalll.getText().toString();
184             if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
185                 textToSpeech.speak(text, TextToSpeech.QUEUE_FLUSH, null, null);
186             } else {
187                 textToSpeech.speak(text, TextToSpeech.QUEUE_FLUSH, null);
188             }
189         }
190     });

```

Gambar 10. DatabaseHelper.java

```

1 package com.mya.drawerlayout;
2
3 import android.app.AlertDialog;
4 import android.content.ActivityNotFoundException;
5 import android.content.Context;
6 import android.content.DialogInterface;
7 import android.content.Intent;
8 import android.database.Cursor;
9 import android.database.sqlite.SQLiteDatabase;
10 import android.net.Uri;
11 import android.os.Build;
12 import android.os.Bundle;
13 import android.speech.RecognizerIntent;
14 import android.speech.tts.TextToSpeech;
15 import android.support.annotation.Nullable;
16 import android.support.v4.app.Fragment;
17 import android.text.Editable;
18 import android.text.TextWatcher;
19 import android.view.LayoutInflater;
20 import android.view.View;
21 import android.view.ViewGroup;
22 import android.widget.Button;
23 import android.widget.EditText;
24 import android.widget.ImageButton;
25 import android.widget.TextView;
26 import android.widget.Toast;
27
28 import java.util.ArrayList;
29 import java.util.Locale;
30
31
32 public class dashboard extends Fragment {
33
34     private SQLiteDatabase db = null;
35     private Cursor kamausCursor = null;
36     private EditText EditTextIndonesia;
37     private TextView txtKalll;
38     private Button btnTerjemahan;

```

```

154 EditTextIndonesia.addTextChangedListener(new dashboard.MyTextWatcher(EditTextIndonesia));
155
156 btnTerjemahan.setOnClickListener(new View.OnClickListener() {
157     @Override
158     public void onClick(View v) {
159         getTerjemahan(v);
160     }
161 });
162
163 btnSpeechToText.setOnClickListener(new View.OnClickListener() {
164     @Override
165     public void onClick(View view) {
166         tanyaInputSuara();
167     }
168 });
169
170 textToSpeech = new TextToSpeech(getActivity(), new TextToSpeech.OnInitListener() {
171     @Override
172     public void onInit(int status) {
173         //method untuk mendeteksi suara dari text
174         if (status == TextToSpeech.ERROR) {
175             textToSpeech.setLanguage(new Locale("id", "ID"));
176         }
177     }
178 });
179
180 btnTextToSpeech.setOnClickListener(new View.OnClickListener() {
181     @Override
182     public void onClick(View view) {
183         String text = txtKalll.getText().toString();
184         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
185             textToSpeech.speak(text, TextToSpeech.QUEUE_FLUSH, null, null);
186         } else {
187             textToSpeech.speak(text, TextToSpeech.QUEUE_FLUSH, null);
188         }
189     }
190 });

```

```

191     }
192     });
193 }
194
195 btnImageClear.setOnClickListener(new View.OnClickListener() {
196     @Override
197     public void onClick(View view) {
198         EditTextIndonesia.setText("");
199         txtKali.setVisibility(View.INVISIBLE);
200         btnTts.setVisibility(View.INVISIBLE);
201         btnImageClear.setVisibility(View.INVISIBLE);
202     }
203 }
204
205
206
207
208
209
210 private class MyTextWatcher implements TextWatcher {
211     private View view;
212
213     private MyTextWatcher(View view) {
214         this.view = view;
215     }
216
217     public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {
218     }
219
220     public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
221         if(charSequence.length() > 0){
222             btnImageClear.setVisibility(View.VISIBLE);
223         }
224         else{
225             btnImageClear.setVisibility(View.INVISIBLE);
226         }
227     }
228
229
230     @Override
231     public void afterTextChanged(Editable editable) {
232     }
233 }
234
235
236
237
238 @Override
239 public void onActivityResult(int requestCode, int resultCode, Intent data) {
240     super.onActivityResult(requestCode, resultCode, data);
241     Toast.makeText(getActivity(), "Hasil suara ditampikan", Toast.LENGTH_SHORT).show();
242     switch (requestCode) {
243         case REQ_CODE_SPEECH_INPUT: {
244             if (resultCode == getActivity().RESULT_OK && null != data) {
245                 ArrayList<String> result = data
246                     .getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
247                 EditTextIndonesia.setText(result.get(0));
248                 getTerjemahan(getView());
249                 break;
250             }
251         }
252     }
253 }
254
255
256
257 public dashboard() {
258     // Required empty public constructor
259 }
260
261 /**
262  * Use this factory method to create a new instance of
263  * this fragment using the provided parameters.
264  *
265  * @param param1 Parameter 1.
266  * @param param2 Parameter 2.
267  *
268  * @return A new instance of fragment dashboard.
269  */
270 // TODO: Rename parameter arguments, choose names that match
271 // the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
272 private static final String ARG_PARAM1 = "param1";
273 private static final String ARG_PARAM2 = "param2";
274
275 // TODO: Rename and change types of parameters
276 private String mParam1;
277 private String mParam2;
278
279 private OnFragmentInteractionListener mListener;
280
281 // TODO: Rename and change types and number of parameters
282 public static dashboard newInstance(String param1, String param2) {
283     dashboard fragment = new dashboard();
284     Bundle args = new Bundle();
285     args.putString(ARG_PARAM1, param1);
286     args.putString(ARG_PARAM2, param2);
287     fragment.setArguments(args);
288     return fragment;
289 }
290
291 @Override
292 public void onCreate(Bundle savedInstanceState) {
293     super.onCreate(savedInstanceState);
294     if (getArguments() != null) {
295         mParam1 = getArguments().getString(ARG_PARAM1);
296         mParam2 = getArguments().getString(ARG_PARAM2);
297     }
298 }
299
300 @Override
301 public void onCreateView(LayoutInflater inflater, ViewGroup container,
302     Bundle savedInstanceState) {
303     // Inflate the layout for this fragment
304     return inflater.inflate(R.layout.fragment_dashboard, container, false);
305 }
306
307 // TODO: Rename method, update argument and hook method into UI event
308 public void onButtonPressed(Uri uri) {
309     if (mListener != null) {
310         mListener.onFragmentInteraction(uri);
311     }
312 }
313
314 @Override
315 public void onAttach(Context context) {
316     super.onAttach(context);
317 }
318
319 @Override
320 public void onDetach() {
321     super.onDetach();
322     mListener = null;
323     try {
324         kamusCursor.close();
325         db.close();
326     } catch (Exception e) {}
327 }
328
329
330 /**
331  * This interface must be implemented by activities that contain this
332  * fragment to allow an interaction in this fragment to be communicated
333  * to the activity and potentially other fragments contained in that
334  * activity.
335  *
336  * See the Android Training lesson <a href=
337  * "https://developer.android.com/training/basics/fragments/communicating.html">
338  * <https://developer.android.com/training/basics/fragments/communicating.html">
339  * </a> for more information.
340  */
341 public interface OnFragmentInteractionListener {
342     // TODO: Update argument type and name
343     void onFragmentInteraction(Uri uri);

```

```

TambahKata.java
1 package com.uyu.drawerlayout;
2
3 import android.app.AlertDialog;
4 import android.app.ProgressDialog;
5 import android.content.ContentValues;
6 import android.content.Context;
7 import android.content.DialogInterface;
8 import android.database.Cursor;
9 import android.database.sqlite.SQLiteDatabase;
10 import android.net.Uri;
11 import android.os.Bundle;
12 import android.support.annotation.Nullable;
13 import android.support.design.widget.TextInputLayout;
14 import android.support.v4.app.Fragment;
15 import android.support.v7.app.AlertDialog;
16 import android.text.Editable;
17 import android.text.TextWatcher;
18 import android.view.LayoutInflater;
19 import android.view.View;
20 import android.view.ViewGroup;
21 import android.view.WindowManager;
22 import android.widget.Button;
23 import android.widget.EditText;
24 import android.widget.ProgressBar;
25 import android.widget.Toast;
26
27 import org.w3c.dom.Text;
28
29
30 public class TambahKata extends Fragment {
31     private SQLiteDatabase db = null;
32     private DatabaseHelper dataKamus = null;
33     private EditText txtIndonesia;
34     private EditText txtKali;
35     private TextInputLayout layoutTxtIndonesia;
36     private TextInputLayout layoutTxtKali;
37     private Button tambahKata;
38     public static final String INDONESIA = "Indonesia";
39     public static final String KALI = "kali";
40
41     private Cursor kamusCursor = null;
42
43     public void saveData(View view) {
44         String khsIndonesia = txtIndonesia.getText().toString();
45         String khsKali = txtKali.getText().toString();
46
47         ContentValues cv = new ContentValues();
48         cv.put(INDONESIA, khsIndonesia.toLowerCase());
49         cv.put(KALI, khsKali.toLowerCase());
50         if (db.insert("Kamus", null, cv) > 0) {
51             Toast.makeText(getActivity().getBaseContext(), "Save Data Success", Toast.LENGTH_SHORT).show();
52             urutDataKamus(view);
53         }
54         else {
55             android.app.AlertDialog.Builder builder1 = new android.app.AlertDialog.Builder(getActivity());
56             builder1.setMessage("Save Data Fail");
57             builder1.setCancelable(true);
58
59             builder1.setPositiveButton(
60                 "OK",
61                 new DialogInterface.OnClickListener() {
62                     public void onClick(DialogInterface dialog, int id) {
63                         dialog.cancel();
64                     }
65                 });
66
67             android.app.AlertDialog alert1 = builder1.create();
68             alert1.show();
69
70         }
71     }
72
73
74
75     public void urutDataKamus(View view) {
76
77         dataKamus = new DatabaseHelper(getActivity().getBaseContext());
78         dataKamus.urutData(dataKamus.getWritableDatabase());
79
80         txtIndonesia.setText("");
81         txtIndonesia.clearFocus();
82         txtKali.setText("");
83         txtKali.clearFocus();
84         layoutTxtIndonesia.setError(null);
85         layoutTxtKali.setError(null);
86
87     }
88
89
90
91
92 private ProgressBar mProgressBar;
93
94 @Override
95 public void onCreateView(View view, @Nullable Bundle savedInstanceState) {
96     super.onCreateView(view, savedInstanceState);
97     dataKamus = new DatabaseHelper(getActivity().getBaseContext());
98     db = dataKamus.getWritableDatabase();
99     txtIndonesia = (EditText) getActivity().findViewById(R.id.input_indonesia);
100     txtKali = (EditText) getActivity().findViewById(R.id.input_kali);
101     layoutTxtIndonesia = (TextInputLayout) getActivity().findViewById(R.id.input_layout_indonesia);
102     layoutTxtKali = (TextInputLayout) getActivity().findViewById(R.id.input_layout_kali);
103     tambahKata = (Button) getActivity().findViewById(R.id.btn_tambahKata);
104
105     txtIndonesia.addTextChangedListener(new MyTextWatcher(txtIndonesia));
106     txtKali.addTextChangedListener(new MyTextWatcher(txtKali));
107
108     tambahKata.setOnClickListener(new View.OnClickListener() {
109         @Override
110         public void onClick(View view) {
111             submitForm(view);
112         }
113     });
114 }
115
116
117 private void submitForm(View view) {
118     if (validateIndonesia()) {
119         return;
120     }
121     if (validateKali()) {
122         return;
123     }
124     saveData(view);
125
126 }
127
128 private void requestFocus(View view) {
129     if (view.requestFocus()) {
130         getActivity().getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_VISIBLE);
131     }
132 }
133
134
135 private boolean validateIndonesia() {
136     if (txtIndonesia.getText().toString().trim().isEmpty()) {
137         layoutTxtIndonesia.setError(getString(R.string.err_msg_indonesia));
138         requestFocus(txtIndonesia);
139         return false;
140     } else {
141         layoutTxtIndonesia.setErrorEnabled(false);
142     }
143     return true;
144 }
145
146
147 private boolean validateKali() {
148     if (txtKali.getText().toString().trim().isEmpty()) {
149         layoutTxtKali.setError(getString(R.string.err_msg_kali));
150         requestFocus(txtKali);
151         return false;
152     } else {

```

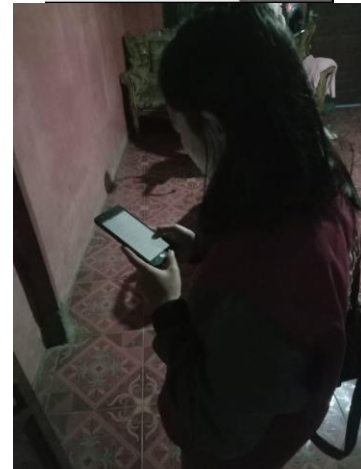
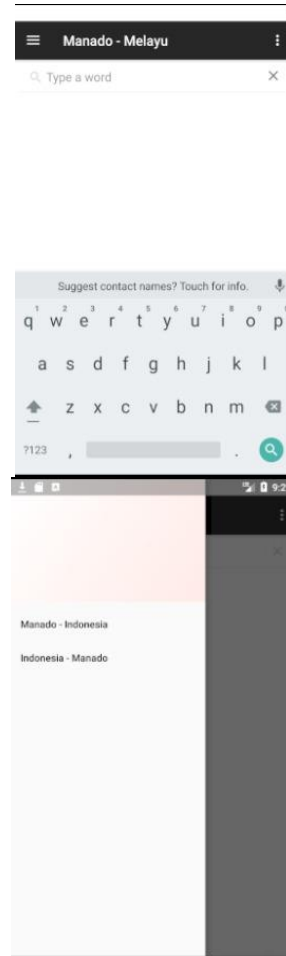
Gambar 11. *Dashboard.java*

```

153         layoutTextKallii.setErrorEnabled(false);
154     }
155     return true;
156 }
157 }
158
159 private class MyTextWatcher implements TextWatcher {
160
161     private View view;
162
163     private MyTextWatcher(View view) {
164         this.view = view;
165     }
166
167     public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {
168     }
169
170     public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
171     }
172
173     @Override
174     public void afterTextChanged(Editable editable) {
175         switch (view.getId()) {
176             case R.id.input_indonesia:
177                 validateIndonesia();
178                 break;
179             case R.id.input_kallii:
180                 validateKallii();
181                 break;
182         }
183     }
184 }
185
186 }
187
188 private OnFragmentInteractionListener mListener;
189
190 public class TambahKata() {
191     // Required empty public constructor
192 }
193
194 @Override
195 public void onCreate(Bundle savedInstanceState) {
196     super.onCreate(savedInstanceState);
197 }
198
199 @Override
200 public View onCreateView(LayoutInflater inflater, ViewGroup container,
201     Bundle savedInstanceState) {
202     // Inflate the layout for this fragment
203     return inflater.inflate(R.layout.fragment_tambah_kata, container, false);
204 }
205
206 @Override
207 public void onStart() {
208     super.onStart();
209 }
210
211 @Override
212 public void onStop() {
213     super.onStop();
214 }
215
216 @Override
217 public void onDestroy() {
218     mListener = null;
219     db.close();
220 }
221
222 /**
223 * This interface must be implemented by activities that contain this
224 * fragment to allow an interaction in this fragment to be communicated
225 * to the activity and potentially other fragments contained in that
226 * activity.
227 * See the Android Training lesson <a href=
228 * "http://developer.android.com/training/basics/fragments/communicating.html"
229 * >Communicating with Other Fragments</a> for more information.
230 */
231 public interface OnFragmentInteractionListener {
232     // TODO: Update argument type and name
233     void onFragmentInteraction(Uri uri);
234 }
235 }

```

Gambar 12. tambahkata.java



Gambar 13. Pengujian Aplikasi

4)Testing

V.KESIMPULAN DAN SARAN

Kesimpulannya adalah Bahasa sangatlah penting dan sangat dibutuhkan oleh masyarakat. Maka dapat disimpulkan bahwa aplikasi kamus Idiom bahasa Manado-Melayu yang berbasis android ini sudah dapat digunakan, aplikasi ini dibangun agar mempermudah para wisatawan maupun anak muda zaman sekarang untuk mengenal Idiom bahasa Manado-

Melayu. Dan juga aplikasi ini dapat membantu untuk memperkenalkan bahasa daerah manado kepada wisatawan.

Saran yaitu Berdasarkan hasil dan pembahasan maka, Aplikasi kamus Idiom bahasa Manado-Melayu yang telah saya buat, dipat diajukan saran sebagai berikut :

1. Untuk pengembangan selanjutnya bisa ditambahkan fitur untuk menambahkan kata
2. Tampilannya lebih bagus lagi
3. Data dapat diperbanyak
4. Dapat dikembangkan untuk IOS dan Web.

VI. KUTIPAN (TNR 8)

- [1] Biber, Douglas, Conrad S, and Leech.G. (2002). *Longman Student Grammar of Spoken and Written English*. England : Pearson Education Limited
- [2] Cowie, A. P. & Mackin, R. (1977) *Offord Dictionary of Current Idiomatic English Vol. 1 : Verb With Prepositions and Particles*
- [3] Liontas, J. I. (2002). *Context and idiom understanding in second languages* In S. Foster-Cohen, T. Ruthenberg, & M. Poschen (Eds.).
- [4] Paramarta, B.P. (2018). Analisa Korpus terhadap Idioms Bahasa Indonesia yang berbasis Nama Binatang. *LINGUAL Jurnal Bahasa dan Sastra dan Pengajaran Fakultas Bahasa dan Seni Universitas Semarang* Vol. XIV, No.1
- [5] Pakasi J. H. (2012). *Dendang Anak Manado di Masa Natal*. Analisis Morfem dan Makna Budaya. *Emolinguistik Minahasa* Sebuah Antologi. Kanisius Yogyakarta.
- [6] P. D. Roger S. Pressman, “Rekayasa Perangkat Lunak - Buku Satu, Pendekatan Praktisi,” in *Software Engineering: A Practitioner’s Approach, Seventh Edition*, 2012
- [7] Yudhanto, Y. and Wijayanto, A. 2017. *Mudah Membuat Dan Berbisnis Aplikasi Android Dengan Android Studio*. PT Elex Media Komputindo. Jakarta
- [8] Yusun (2018). ‘Translation Strategies Applied in English and Chinese Idioms’. *Advances in Social Science, Education and Humanities Research (ASSEHR)*, vol. 300 2018 International Workshop on Education Reform and Social Sciences (ERSS). Atlantis Press

Johnny Wuwungan dan Nouce Susane Tewuh. Penulis mulai menempuh pendidikan di Sekolah Dasar Santa Theresia Manado (2003-2009). Penulis lalu melanjutkan ke Sekolah Menengah Pertama Negeri 8 Manado (2009-2012). Kemudian penulis melanjutkan ke Sekolah Menengah Kejuruan Negeri 1 Manado (2012-2015). Pada tahun 2015 penulis melanjutkan Pendidikan ke perguruan tinggi yang ada di Manado yaitu Universitas Sam Ratulangi Manado, dengan mengambil Program Studi S-1 Teknik Informatika di Jurusan Teknik Elektro Fakultas Teknik. Pada bulan Februari tahun 2020 Penulis mengajukan proposal Skripsi untuk memenuhi syarat meraih gelar sarjana (S1) dengan judul Aplikasi Android Kamus Idiom Bahasa Manado-Melayu yang kemudian disetujui dan melanjutkan pembuatan penelitian skripsi. Pembuatan skripsi ini dibimbing oleh dua dosen pembimbing, yaitu Yaulie D. Y. Rindengan, ST., MM., MSc. dan Xaverius B. N. Najoan, ST., MT.



Andrew Wuwungan, lahir di Bekasi 16 Desember 1997. Penulis merupakan anak ke-2 dari 3 bersaudara, dari pasangan