

Perancangan Kamus Tuna Rungu dengan Fitur *ArrayAdapter* dan Metode *TextWatcher* berbasis *Android Mobile*

Lukman Hakim¹⁾, Maily Jaya Suharja²⁾

Teknik Informatika Universitas Bunda Mulia

JL. Lodan Raya No.2 Jakarta 14430, (021) -6929090 ext 617

e-mail: lhakim@bundamulia.ac.id¹⁾, maildy_coolzz@yahoo.com²⁾

Abstrak

Kamus pengenalan bahasa tuna rungu berbasis android merupakan suatu cara untuk memperkenalkan kata, angka, huruf dalam bentuk bahasa isyarat tangan kepada tuna rungu yang digunakan untuk belajar bahasa dalam berkomunikasi setiap hari. Kamus tersebut yang dijalankan pada telepon genggam cerdas atau *Smartphone* memberikan nilai lebih dalam belajar bahasa tuna rungu secara visual berdasarkan gambar yang disesuaikan dengan bahasa Indonesia. Perancangan aplikasi ini menggunakan fitur *ArrayAdapter* dan method *TextWatcher* adalah method yang bisa digunakan pada aplikasi mobile berbasis android. fitur *ArrayAdapter* dan method *TextWatcher* merupakan method pencarian yang terdapat pada package *WidgetAdapter* yang berfungsi untuk melakukan pencarian yang akurat. *Adapter* memiliki caranya sendiri untuk memfilter data dan *TextWatcher* merupakan salah satu jenis *filtering data* pada *ArrayAdapter* yang digunakan untuk pencarian huruf, angka, kata melalui *EditView*. Kamus pengenalan bahasa tuna rungu dapat dijalankan pada *Android* minimal versi 2.3 (*gingerbread*) keatas dengan resolusi 240x400 pixel pada telpn genggam.

Kata kunci: Tuna rungu, *TextWatcher*, *ArrayAdapter*.

1. Pendahuluan

Berdasarkan radio berita Australia (ABC news, 2014) Indonesia merupakan negara terbesar ke 4 didunia, setiap tahun penderita gangguan pendengaran atau Tuna rungu mengalami kenaikan sehingga 5000 penderita. Hal tersebut diakui oleh badan kesehatan dunia (WHO) satu dari seribu bayi yang lahir didunia mengalami tuli atau gangguan pendengaran (Momdadi, 2013), gangguan pendengaran ini disebabkan oleh yaitu:terinfeksi virus toksoplasmosis, herpes, sipilis, gen keturunan yang diwariskan. Lalu dengan kondisi seperti ini anak-anak tersebut perlu penanganan sehingga mereka dapat berinteraksi dilingkungan sekitarnya. Penanganan penderita tuna rungu banyak menggunakan bermacam-macam cara baik dengan menggunakan alat bantu dengar atau implantasi koklea (rumah siput) atau dengan sekolah khusus yang mempelajari gerak tangan sebagai media komunikasi mereka[1].

penyandang tunarungu bisa jadi tidak jauh berbeda dengan manusia pada umumnya. Mereka tidak memperlihatkan perbedaan yang kasat mata seperti penyandang tunadaksa. Perbedaan mereka terletak pada kemampuan komunikasi yang dipengaruhi kemampuan menangkap suara dan menyuarakan kata. Kemampuan komunikasi yang terbatas membuat para tuna rungu susah berkomunikasi serta sarana pembelajaran bahasa insyarat mereka tidak semua dapat ditemukan atau kamus tuna rungu dijual pada setiap toko buku. Kenyataan inilah membuat terjadinya perbedaan yang sangat jauh dengan orang normal serta kebijakan pemerintah yang belum sepenuhnya memperhatikan hak mereka sebagai warga negara. Hal tersebut mendorong dibuatnya kamus tuna rungu berbasis *mobile* serta perkembangan teknologi komunikasi seperti telepon genggam, memberikan kemudahan tersendiri pada semua orang menjadikan telpn genggam sebagai perangkat yang serba guna seperti aplikasi pembelajaran, *game*, kamus, dan seterusnya. Aplikasi Kamus tuna rungu berbasis *mobile* yang menampilkan gambar isyarat tangan yang digunakan untuk komunikasi para tuna rungu menjadi satu solusi untuk digunakan sebagai media pembelajaran bahasa isyarat yang digunakan untuk bersosialisasi dengan masyarakat.

Permasalahan pada penelitian ini adalah bagaimana fitur *ArrayAdapter* dan metode *TextWatcher* digunakan untuk pencarian kata dan suku pada perancangan kamus tuna rungu berbasis *mobile*.

Manfaat kamus tuna rungu memberikan kemudahan dalam pengenalan kata, suku kata atau angka dan abjad bahasa isyarat yang digunakan sehari-hari.

2. Metode Penelitian

Data dan referensi diambil dari beberapa buku yang berhubungan dengan tuna rungu dan pemrograman Android dan xml, membuat kode java pada eclipse dan melakukan wawancara dengan pengelola sekolah berkebutuhan khusus tuna rungu di Panti Sosial Bina Rungu Wicara “MELATI”-Jakarta Selatan .

2.1. Fitur *ArrayAdapter* dan Metode *TextWatcher*.

2.1.1. Fitur *ArrayAdapter*

ArrayAdapter adalah salah satu kelas yang terdapat pada *package Widget Adapter* yaitu *ListAdapter* yang mengelola *object* yang berbentuk *array*. Untuk menampilkan data pada *ListView*, umumnya data dimasukkan ke dalam *Adapter* yang kemudian ditampilkan ke dalam *ListView*. *Adapter* bisa juga dianalogikan sebagai model dalam konsep MVC (*Model View Controller*) . Ada berbagai macam *Adapter* yang bisa digunakan sesuai dengan kebutuhan tertentu. Salah satunya adalah *ArrayAdapter* yang ditujukan untuk menampilkan hasil *query* dari basisdata ke dalam *ListView*, terlihat pada deklarasi *code* dibawah ini[2].

```
adapter = new ArrayAdapter<String>(this,
    android.R.layout.simple_expandable_list_item_1, listKamus);
lv.setAdapter(adapter);
```

2.1.2. Metode *TextWatcher*

Masing-masing jenis *Adapter* memiliki caranya sendiri untuk memfilter data. *TextWatcher* merupakan salah satu jenis filtering data pada *ArrayAdapter*. Method *TextWatcher* akan memfilter setiap huruf yang dimasukkan atau dihapus melalui *EditView* dan memasukkan data ke dalam *Adapter* yang kemudian langsung ditampilkan kembali ke dalam *ListView*. Terlihat pada potongan method dibawah ini[3].

```
public void afterTextChanged(Editable arg0) {
    @Override
    public void beforeTextChanged(CharSequence arg0, int arg1, int arg2, int arg3) {}
    @Override
    public void onTextChanged(CharSequence s, int arg1, int arg2, int arg3) {
        adapter.getFilter().filter(s.toString());
    }
}
```

2.2. SQLite Database

Android memiliki fasilitas untuk membuat basis data yang dengan *SQLite*. *SQLite* adalah salah satu perangkat lunak yang *embedded* yang sangat populer, kombinasi *SQL interface* dan penggunaan memori yang sangat sedikit dengan kecepatan yang sangat cepat. *SQLite* di *Android* termasuk dalam *Android runtime*, sehingga setiap versi dari *Android* dapat membuat basis data dengan *SQLite*. Pengembang hanya cukup mendeklarasikan perintah *SQL* untuk membuat atau memperbaharui basis data, selanjutnya sistem pada *Android* akan menangani hal-hal yang berhubungan dengan basis data. Basis data *SQLite* otomatis akan tersimpan di dalam *path* data/data/nama_package/database/nama_database[4].

2.3. Arsitektur Android

Mulyadi (2010) menjelaskan bahwa arsitektur android terdiri dari:Linux Kernel, Libraries, Android- Runtime, Framework-aplikasi dan Applications. Android telah menyertakan aplikasi inti seperti email client, sms, kalender, peta, browser, kontak, dan lain-nya[5].

2.4. Komponen Android

Menurut Mulyadi(2010) Android merupakan *subset* perangkat lunak untuk perangkat *mobile* yang meliputi sistem operasi, *middleware* dan aplikasi ini yang di rilis oleh google, sedangkan android SDK (*Software Development Kit*) menyediakan tools dan API yang diperlukan untuk mengembangkan aplikasi pada *platform* Android dengan menggunakan bahasa pemrograman java[5].

2.5. XML Android

Menurut Huda (2014) *XML Android* pada dasarnya berbeda dengan *XML* secara umum. Pada *XMLAndroid*, nama-nama tag telah didefinisikan oleh *Android SDK*, tidak seperti *XML* pada umumnya yang dapat menuliskan nama-nama tag sesuai keinginan. Selain nama-nama tag, pada *XML Android* juga menyediakan perintah untuk mengatur *widget-widget* tersebut yang dituliskan di dalam tag-tag tersebut,

semua widget XML yang digunakan nantinya akan masuk atau otomatis diatur pada *class* R.java yang mana setiap *widget* dikelompokkan berdasarkan jenisnya seperti, *id*, *layout*, *text*, dan lain-lain.[6]

3. Hasil dan Pembahasan

3.1. Tampilan Halaman tunggu Kamus Tuna Rungu “Bebas”

Aplikasi *Android* menggunakan bahasa *XML* yang berfungsi untuk mengatur antarmuka tampilan aplikasi, seperti mengatur letak posisi tombol, gambar latar belakang, posisi teks dan hampir semua yang berhubungan dengan tampilan. *XML* juga digunakan untuk manajemen *resource*, membuat animasi, dan banyak lainnya. Tampilan pada halaman tunggu kamus tuna rungu pada gambar 1.



Gambar 1. Tampilan Halaman tunggu kamus tuna rungu

3.2. Tampilan Halaman Utama Kamus Tuna Rungu

Pada halaman menu utama terdapat 4 bagian yang berbeda sebagai dasar antarmuka menu utama. Pada setiap Bagian terdiri dari *header*, *footer*, *fragment_layout*, dan 4 buah tombol. Tampilan menu utama pada gambar 2.



Gambar 2. Tampilan Halaman Utama kamus Tuna Rungu

3.3. Tampilan halaman Abjad Kamus Tuna Rungu

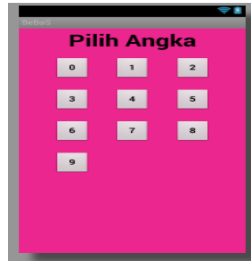
Pada halaman ini sistem menampilkan daftar abjad yang terdapat pada *database*. Pada halaman ini pengguna dapat memilih abjad yang diinginkan dengan cara menekan tombol abjad yang tersedia pada layar. Tampilan halaman abjad pada gambar 3.



Gambar 3. Tampilan Halaman Abjad kamus Tuna Rungu

3.4. Tampilan halaman angka Kamus Tuna Rungu

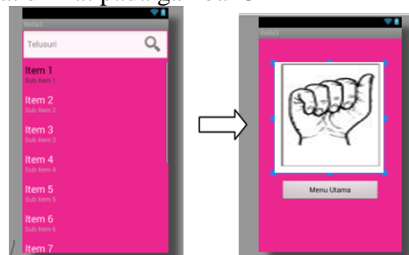
Pada halaman ini sistem menampilkan daftar angka 0-9 yang terdapat pada *database*. Pada halaman ini pengguna dapat memilih angka yang diinginkan dengan cara menekan tombol angka yang tersedia pada layar. Tampilan halaman abjad dapat dilihat pada gambar 4.



Gambar 4 Tampilan Halaman angka kamus Tuna Rungu

3.5. Tampilan halaman pencarian kata Kamus Tuna Rungu dengan metode *TextWatcher*

Halaman pencarian atau daftar kata merupakan fitur yang dimiliki kamus Tuna Rungu untuk memberikan kemudahan dalam melakukan pencarian setiap kata yang sudah terdapat dalam *database*, penerapan metode *TextWatcher* digunakan untuk melakukan pencarian secara terinci dari setiap huruf pada kata. Tampilan halaman pencarian dapat dilihat pada gambar 5

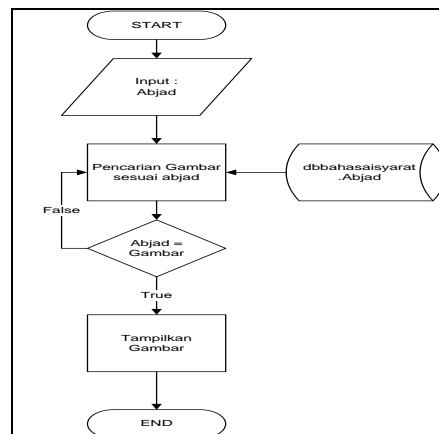


Gambar 5. Tampilan Halaman pencarian kata kamus Tuna Rungu

3.5. Pembahasan

3.5.1. Flowchar Pencarian abjad

Pada saat melakukan pencarian baik berupa abjad yang ditampilkan dalam *ArrayAdapter* melakukan pencarian berdasarkan kesesuaian data yang diminta, apabila ditemukan data yang diminta setelah itu data yang diminta akan ditampilkan oleh *ArrayAdapter* berupa gambar yang sudah disimpan. Untuk flowchar pencarian abjad dapat dilihat pada gambar 6. sebagai berikut:



Gambar 6. Flowchar pencarian abjad

3.5.2. Koding pencarian kata dengan *TextWatcher*

Penerapan pencarian mencari kata dengan *ListView* dan juga dapat mencari kata melalui *EditText* telusuri. Pada pendeklarasian kelas tersebut melakukan implementasi pada kelas *TextWatcher*, fitur *EditText* melakukan pencarian secara rinci pada setiap huruf pada kata yang berada didalam *database*. Potong code pencarian dapat dilihat pada gambar 7.

```

Public class Pencarian extends Activity implements TextWatcher, OnItemClickListener {
    private EditText search;
    private ListView lv;
    private DBKamusHelper dbHelper;
    private ArrayAdapter<String> adapter;
  
```

```

private List<String> listKamus;
private SQLiteDatabase db = null;
@Override
protected void onCreate(Bundle savedInstanceState) {
    .....
}
private void setData() {
    listKamus = dbHelper.getAllLabels();
    adapter = new ArrayAdapter<String>(this,
    android.R.layout.simple_expandable_list_item_1, listKamus);
    lv.setAdapter(adapter); }
@Override
public void afterTextChanged(Editable arg0) {}
@Override
public void beforeTextChanged(CharSequence arg0, int arg1, int arg2,
    int arg3) {}
@Override
public void onTextChanged(CharSequence s, int arg1, int arg2, int arg3) { adapter.getFilter().filter(s.toString());
}
@Override
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    Bundle b = new Bundle();
    b.putString("jenis", adapter.getItem(position));

    Intent i = new Intent(this, HasilCari.class);
    i.putExtras(b);
    startActivity(i);
}
}

```

Gambar 7. Potongan koding pencarian dengan *TextWatcher*

3.5.3. potongan code hasil pencarian

Pada code ini merupakan kelanjutan dari kelas pencarian pada gambar 7. Kelas hasilcari langsung melakukan sorting kedalam database kamus yang berisi informasi jenis dan image yang sudah disesuaikan berdasarkan dengan penamaan isyarat. Potong code hasilcari dapat dilihat pada gambar 8.

```

public class HasilCari extends Activity {
    private SQLiteDatabase db = null;
    private DBKamusHelper dataDB = null;
    private Cursor cursor = null;
    private String idjenis = null;
    private TextView jenis, imageText;
    private ImageView image;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        .....
        .....
        getKamusDetail(); }
    public void getKamusDetail() {
        Bundle b = getIntent().getExtras();
        idjenis = b.getString("jenis");
        try {
            cursor = db.rawQuery("SELECT * FROM Kamus where jenis = '" + idjenis + "';", null);
            if (cursor.getCount() > 0) {
                int indexJenis = cursor.getColumnIndex("jenis");
                int indexImage = cursor.getColumnIndex("image");
                cursor.moveToFirst();
                do {
                    String Jenis = cursor.getString(indexJenis);
                    String Image = cursor.getString(indexImage);
                    jenis.setText(Jenis);
                    imageText.setText(Image);
                    if (Image.equals("astim.png")) {
                        image.setImageResource(R.drawable.astim);
                    } elseif (Image.equals("bstim.png")) {
                        image.setImageResource(R.drawable.bstim);
                    } elseif (Image.equals("cstim.png")) {
                        image.setImageResource(R.drawable.cstim);
                    } elseif (Image.equals("dstim.png")) {
                        image.setImageResource(R.drawable.dstim);
                    }
                    cursor.moveToNext();
                } while (!cursor.isAfterLast());
            }
        } finally {

```

```
if (cursor != null) {  
    cursor.close();  
}
```

Gambar 8. Potongan koding Hasil cari pada kamus Tuna Rungu.

4. Simpulan

Dari hasil pengujian kamus Tuna rungu disimpulkan sebagai berikut :

1. Penggunaan fitur *ArrayAdapter* dan *TextWatcher* sangat efektif dalam melakukan pencarian kesesuaian huruf atau kata untuk mempermudah proses pencarian lebih cepat. Hal ini terbukti dari beberapa kosa kata yang dimasukan proses pencarian cepat direspon oleh aplikasi kamus tuna rungu.
2. Dapat berjalan dengan baik pada perangkat bergerak berbasis *Android* versi 2.3 keatas (*Gingerbread*) dengan resolusi minimal 240 x 400 *pixels*, seperti *Samsung Galaxy Ace3*, *OPPOR287 Find Five*, *Lenovo A3000 Tablet*, dan *Samsung Galaxy Tab 3*.
3. Aplikasi kamus Tuna Rungu memberikan kemudahan dalam proses pembelajaran pengenalan bahasa isyarat Indonesia, dengan tampilan sederhana memudahkan dalam penggunaan.

Saran

Kamus tuna rungu sebaiknya dalam perkembangan kedepan perlu ditambahkan kosa kata yang lebih lengkap untuk memudahkan perbendaharaan kata dalam komunikasi dengan bahasa isyarat, dan aplikasi dapat berjalan pada operasi sistem selain *Android* seperti *iOS*, *Windows phone*, dari sisi desain interface komposisi pewarnaan yang sesuai dengan 9 aturan Interaksi Manusia dan Komputer.

Daftar Pustaka

- [1] <http://www.radioaustralia.net.au/indonesian/radio/onairhighlights/methode-penangananterbaik-untuk-tuna-rungu>
- [2] <http://developer.android.com/reference/android/widget/ArrayAdapter.html>
- [3] <http://developer.android.com/reference/android/text/TextWatcher.html>
- [4] Parno, dkk, Aplikasi *Mobile* Kamus istilah Psikologi berbasis *Android* 2.2, Proceeding Pesat, Universitas Gunadarma, Depok, 2011.
- [5] Mulyadi, S., Membuat Aplikasi untuk *Android* Multimedia Center Publishing Yogyakarta, 2010
- [6] Huda, Arif Akbarul, *LiveCoding! 9 Aplikasi Android Buatan Sendiri*, Andi Publisher, Yogyakarta, 2014. (XML)
- [7] Safaat, Nazruddin, Pemrograman Aplikasi *Mobile Smartphone* dan *Tablet PC* Berbasis *Android* , Informatika, Bandung, 2011,