

LAMPIRAN

A. Kuesioner Pengujian User

KUESIONER APLIKASI ANDROID PEMBELAJARAN ANATOMI GIGI

Nama :

Jurusan :

1. Apakah aplikasi Pembelajaran Anatomi Gigi mudah digunakan?
 - Sangat setuju
 - Setuju
 - Cukup setuju
 - Tidak setuju
 - Sangat tidak setuju
2. Apakah informasi yang ditampilkan dalam aplikasi Pembelajaran Anatomi Gigi dapat ditangkap dengan mudah dan jelas?
 - Sangat setuju
 - Setuju
 - Cukup setuju
 - Tidak setuju
 - Sangat tidak setuju
3. Apakah aplikasi Pembelajaran Anatomi Gigi memiliki tampilan yang menarik?
 - Sangat setuju
 - Setuju
 - Cukup setuju
 - Tidak setuju
 - Sangat tidak setuju
4. Apakah fitur pencarian pada aplikasi Pembelajaran Anatomi Gigi sangat membantu?
 - Sangat setuju
 - Setuju
 - Cukup setuju

- Tidak setuju
 - Sangat tidak setuju
5. Apakah fitur bookmark pada aplikasi Pembelajaran Anatomi Gigi sangat membantu?
- Sangat setuju
 - Setuju
 - Cukup setuju
 - Tidak setuju
 - Sangat tidak setuju
6. Apakah aplikasi Pembelajaran Anatomi Gigi sangat bermanfaat?
- Sangat setuju
 - Setuju
 - Cukup setuju
 - Tidak setuju
 - Sangat tidak setuju

B. Coding Class Pencarian

```
package deka.toothe;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;

public class SearchActivity extends AppCompatActivity{

    final static String urlAddress =
"https://999anatomi.000webhostapp.com/select_all.php";
    final static String urlAddress_filter =
"https://999anatomi.000webhostapp.com/select_filter.php?nama=";
    final static String urlAddress_filter2 =
"https://999anatomi.000webhostapp.com/filter.php?nama=";
    final static String urlAddress_filter3 =
"https://999anatomi.000webhostapp.com/filter_ilmiah.php?nama=";

    EditText src;
    Button src_bt,src_bt2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_search);

        src = (EditText) findViewById(R.id.search_tx);
        src_bt = (Button) findViewById(R.id.search_bt);
        src_bt2 = (Button) findViewById(R.id.search_bt2);

        src.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start,
```

```

    int count, int after) {

        }

        @Override
        public void onTextChanged(CharSequence s, int start, int
before, int count) {

        }

        @Override
        public void afterTextChanged(Editable s) {

            //final DataObject dataObject = new DataObject();
            //final ListView listView = (ListView)
findViewById(R.id.lv1);
            //new Downloader(SearchActivity.this,
urlAddress_filter2 + src.getText().toString(),listView ).execute();
        }
    });

    src_bt.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            hideSoftKeyboard();
            final DataObject dataObject = new DataObject();
            final ListView listView = (ListView)
findViewById(R.id.lv1);

            new Downloader(SearchActivity.this, urlAddress_filter
+ src.getText().toString(),listView ).execute();
        }
    });

    src_bt2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            hideSoftKeyboard();
            final DataObject dataObject = new DataObject();
            final ListView listView = (ListView)
findViewById(R.id.lv1);

```

```

        new Downloader(SearchActivity.this, urlAddress_filter3
+ src.getText().toString(), listView ).execute();
    }
    });
}

/**
 * Menghilangkan keyboard
 */
public void hideSoftKeyboard() {
    if(getCurrentFocus() !=null) {
        InputMethodManager inputMethodManager =
(InputMethodManager) getSystemService(INPUT_METHOD_SERVICE);

inputMethodManager.hideSoftInputFromWindow(getCurrentFocus().getWindow
Token(), 0);
    }
}
}

```

C. Coding Class Bookmark

```

package deka.toothe;

import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.RecyclerView;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class BookmarkActivity extends AppCompatActivity {

    ListView listView;

```

```

AlertDialog.Builder dialog;
List<FavoritObject> itemlist = new ArrayList<FavoritObject>();
FavoritAdapter favoritAdapter;
DbHelper SQLite = new DbHelper(this);

public static final String TAG_ID = "id";
public static final String TAG_NAMA = "nama";
public static final String TAG_NAMA_ILMIAH = "nama_ilmiah";
public static final String TAG_GAMBAR = "gambar";

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_bookmark);

    SQLite = new DbHelper(getApplicationContext());
    listView = (ListView) findViewById(R.id.lv1);

    ArrayList<HashMap<String, String>> favList =
SQLite.getAllData();

    favoritAdapter = new FavoritAdapter(BookmarkActivity.this,
itemlist);
    listView.setAdapter(favoritAdapter);

    listView.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
        @Override
        public boolean onItemLongClick(AdapterView<?> parent, View
view, int position, long id) {
            final String idx = itemlist.get(position).getId();

            final CharSequence[] dialogitem = {"Hapus"};
            dialog = new
AlertDialog.Builder(BookmarkActivity.this);
            dialog.setCancelable(true);
            dialog.setItems(dialogitem, new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int
which) {

                    switch (which) {
                        case 0:
                            SQLite.delete(Integer.parseInt(idx));
                            itemlist.clear();
                            getAllData();
                            break;

                        }
                    }
                }).show();
                return true;
            }
        });

    listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view,
int position, long id) {

```

```

        final String idx = itemList.get(position).getId();
        final String namax = itemList.get(position).getNama();
        final String nama_ilmiahx =
itemlist.get(position).getNama_ilmiah();
        final String gambarx =
itemlist.get(position).getGambar();

        Intent intent = new Intent(BookmarkActivity.this,
DetailsFavorit.class);
        intent.putExtra(TAG_ID, idx);
        intent.putExtra(TAG_NAMA, namax);
        intent.putExtra(TAG_NAMA_ILMIAH, nama_ilmiahx);
        intent.putExtra(TAG_GAMBAR, gambarx);
        startActivity(intent);
    }
});

getAllData();
}

private void getAllData(){
    ArrayList<HashMap<String, String>> row = SQLite.getAllData();

    for (int i = 0; i < row.size(); i++){
        String id = row.get(i).get(TAG_ID);
        String nama = row.get(i).get(TAG_NAMA);
        String nama_ilmiah = row.get(i).get(TAG_NAMA_ILMIAH);
        String gambar = row.get(i).get(TAG_GAMBAR);

        FavoritObject dataObject = new FavoritObject();

        dataObject.setId(id);
        dataObject.setNama(nama);
        dataObject.setNama_ilmiah(nama_ilmiah);
        dataObject.setGambar(gambar);

        itemList.add(dataObject);
    }
    favoritAdapter.notifyDataSetChanged();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {

    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.delete_all_menu, menu);
    return true;
}

public boolean onOptionsItemSelected(MenuItem item) {

    if (item.getItemId() == R.id.deleteall){

        AlertDialog.Builder dialog = new
AlertDialog.Builder(BookmarkActivity.this);
        dialog.setCancelable(true);
        dialog.setMessage("Hapus Semua Bookmark");
        dialog.setPositiveButton("Ya", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which)
            {

                SQLite.deleteAll();
                itemList.clear();
            }
        });
    }
}

```

```

        getAllData();
    }
});
dialog.setNegativeButton("Tidak", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which)
    {
        dialog.dismiss();
    }
});
AlertDialog alert = dialog.create();
alert.show();
}
return true;
}
}

```