

CARA PENGOPERASIAN (MANUAL) Mobile e-Learning Itenas (MeLI)

Berikut ini adalah tata cara/langkah-langkah dalam mengoperasikan aplikasi MeLI mulai dari pendaftaran hingga mengikuti quiz.

1. Pendaftaran Pengguna Baru

Untuk menggunakan fasilitas yang ada pada aplikasi MeLI, pengguna wajib melakukan pendaftaran dengan cara memilih menu daftar dan mengisi data yang wajib dimasukkan. Untuk lebih jelasnya, dapat dilihat pada gambar berikut ini.

The image shows a mobile application interface for registration. At the top, there is a status bar with the time 12:37 PM and various icons. Below the status bar is the MeLI logo, which consists of the letters 'meLI' in a stylized orange font. Underneath the logo, the text 'Mobile e-Learning itenas' is displayed. The registration form includes several input fields: 'Nama Depan' (First Name), 'Nama Belakang' (Last Name), 'e-mail', 'Password' (with a toggle icon for visibility), and 'NRP (Tanpa -, cth: 152018XXX)' (NRP (Without -, example: 152018XXX)). There is a character count '0 / 9' next to the NRP field. Below these fields is a dropdown menu labeled '-- Pilih Jurusan --' (Select Program). At the bottom of the form is a yellow button labeled 'DAFTAR' (Register).

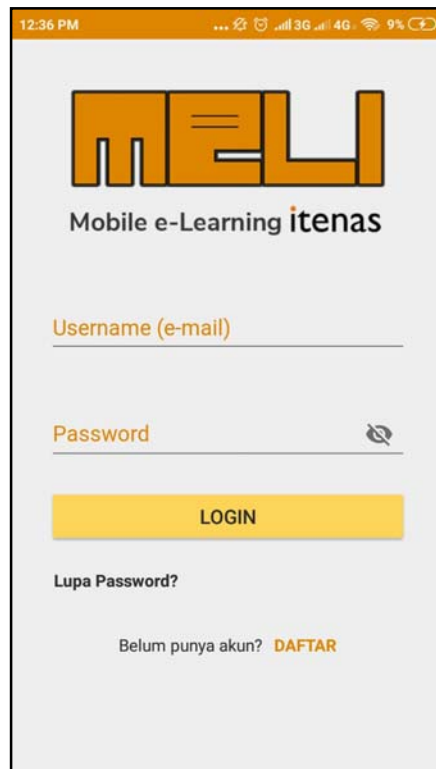
2. Konfirmasi dan Aktivasi akun Pengguna Baru

Setelah melakukan pendaftaran, sistem akan mengirim email ke email pengguna secara otomatis untuk proses aktivasi. Untuk lebih jelasnya, dapat dilihat pada gambar berikut ini.



3. Proses Login

Setelah berhasil melakukan pendaftaran, maka pengguna dapat melakukan proses login pada aplikasi. Untuk lebih jelasnya, dapat dilihat pada gambar berikut ini.



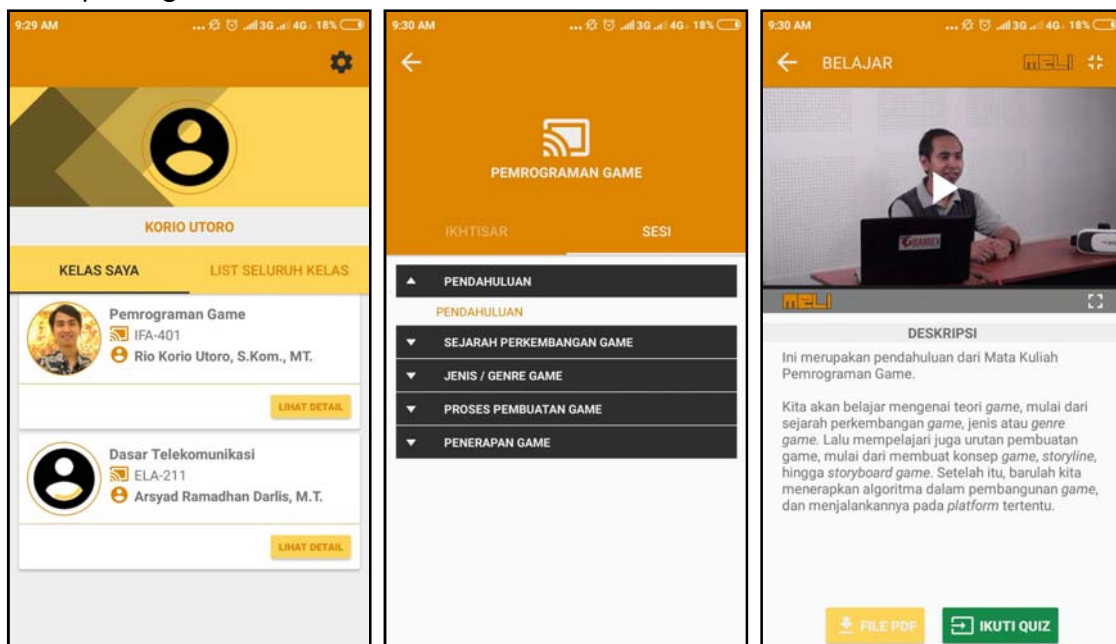
4. Enroll Mata Kuliah

Setelah proses login berhasil dilakukan, pengguna dapat melihat daftar matakuliah yang akan diikuti sesuai dengan jurusanannya. Untuk mengikuti kuliah tersebut, pengguna dapat melakukan proses Enroll Mata Kuliah. Untuk lebih jelasnya, dapat dilihat pada gambar berikut ini.



5. Ikuti Kuliah

Setelah berhasil melakukan pendaftaran Mata Kuliah (enroll), pengguna dapat mengikuti perkuliahan melalui aplikasi e-Learning ini. Untuk lebih jelasnya, dapat dilihat pada gambar berikut ini.



6. Ikuti Quiz

Pada aplikasi ini, pengguna memungkinkan untuk mengikuti quiz yang diselenggarakan oleh dosen mata kuliah. Untuk lebih jelasnya, dapat dilihat pada gambar berikut ini.

The screenshot shows a mobile application interface for a quiz. At the top, there is a status bar with the time 9:30 AM, signal strength, and battery level at 18%. Below the status bar is a navigation bar with a back arrow and the title 'PERSAMAAN KUADRAT'. The main content area has two input fields for 'Kelas' and 'Kode Kelas'. Below these fields is a note: '* Kelas dan Kode Kelas diberikan oleh Dosen. Kosongkan jika Dosen tidak memberikan kode.' The quiz details section includes 'Judul Quiz: PERSAMAAN KUADRAT', 'Deskripsi: Persamaan Kuadrat Deskripsi', 'Durasi: 5 menit', and 'Minimum Nilai: 70'. A green 'MULAI' button with a right arrow is positioned below the details. At the bottom, there is a section titled 'Hasil Sebelumnya' with a refresh icon. This section contains a table with the following data:

NO	WAKTU AKSES	TOTAL SKOR	NILAI	STATUS
1	19-12-2018	2	0	gagal
2	20-12-2018	0	0	gagal

LISTING PROGRAM

Mobile e-Learning Itenas (MeLI)

- **Splash Screen Activity**

```
package com.rku.meli;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.view.View;
import android.view.animation.AlphaAnimation;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;
import android.widget.ProgressBar;
import android.widget.TextView;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.rku.meli.SQLite.Session;

import org.json.JSONException;
import org.json.JSONObject;

/**
 * Created by Korio Utoro on 9/27/2018.
 */

public class SplashScreenActivity extends Activity {

    private Session session;

    ImageView iv1, iv2;
    TextView tvCopyright;
    Animation uptodown, downtoup, fadein;

    ProgressBar progressBar;

    public static int SPLASH_TIME_OUT = 3000;

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

        Handler handler;
        session = new Session(this);

        iv1 = (ImageView) findViewById(R.id.logo_leon);
        iv2 = (ImageView) findViewById(R.id.logo_learning_online);
        tvCopyright = (TextView) findViewById(R.id.tv_copyright);

        progressBar = (ProgressBar) findViewById(R.id.splash_progressbar);
```

```

        uptodown = AnimationUtils.loadAnimation(this, R.anim.uptodown);
        downtoup = AnimationUtils.loadAnimation(this, R.anim.downtoup);
        fadein = new AlphaAnimation(0.0f, 1.0f);
        fadein.setDuration(2000);
        iv1.setAnimation(uptodown);
        iv2.setAnimation(downtoup);
        tvCopyright.startAnimation(fadein);
        if (session.loggedIn())
        {
            String id_login = session.getIdLogin();
            Log.d("ID LOGIN DI SPLASH", id_login);

            final int id_user = Integer.valueOf(id_login);

            handler = new Handler();
            handler.postDelayed(new Runnable() {
                @Override
                public void run() {
                    initUser(id_user);
                }
            }, SPLASH_TIME_OUT);

        }else {
            session.setLoggedIn(false);
            handler = new Handler();
            handler.postDelayed(new Runnable() {
                @Override
                public void run() {
                    startActivity(new Intent(SplashScreenActivity.this,
LoginActivity.class));
                    finish();
                }
            }, SPLASH_TIME_OUT);
        }

    }

    private void initUser(int userID)
    {
        String url_get_user =
"https://meli.voidsolution.com/api/get_user_api/";

        JSONObjectRequest request = new
JSONObjectRequest(Request.Method.GET, url_get_user + "?user_id=" + userID,
null, new Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject response) {

                Log.d("RESPONSE GET USER", response.toString());

                progressBar.setVisibility(View.INVISIBLE);

                try {
                    int status_getUser = response.getInt("status");

                    if (status_getUser == 0)
                    {

                        String msg_getUser = response.getString("msg");

```

```

        new AlertDialog.Builder(SplashScreenActivity.this)
            .setTitle("Oops")
            .setMessage(msg_getUser)
            .setCancelable(false)
            .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface
dialogInterface, int i) {
                    dialogInterface.cancel();
                    startActivity(new
Intent(SplashScreenActivity.this, LoginActivity.class));
                }
            }).show();

    }else
    {

        JSONObject dataResponse =
response.getJSONObject("data");

        Log.d("RESPONSE GETUSER SPLASH",
dataResponse.toString());

        String user_id = dataResponse.getString("id");
        String nama_depan =
dataResponse.getString("namadepan");
        String nama_belakang =
dataResponse.getString("namabelakang");
        String display_nama = nama_depan.toUpperCase() + "
" + nama_belakang.toUpperCase();
        String email = dataResponse.getString("email");
        String phone1 = dataResponse.getString("phone");
        String address = dataResponse.getString("address");
        String city = dataResponse.getString("city");
        String country = dataResponse.getString("country");
        String url_foto =
dataResponse.getString("url_foto");
        String status = dataResponse.getString("status");

        session.setIdLogin(user_id);
        session.setUserName(display_nama);
        session.setEmailUser(email);
        session.setPhoneUser(phone1);
        session.setAddressUser(address);
        session.setCityUser(city);
        session.setCountryUser(country);
        session.setStatusUser(status);

        Intent intent = new
Intent(SplashScreenActivity.this, DashboardActivity.class);
        intent.putExtra("ID", user_id);
        intent.putExtra("NAMA", display_nama);
        intent.putExtra("EMAIL", email);
        intent.putExtra("PHONE", phone1);
        intent.putExtra("ADDRESS", address);
        intent.putExtra("CITY", city);
        intent.putExtra("COUNTRY", country);
        intent.putExtra("FOTO", url_foto);
        intent.putExtra("STATUS", status);
    }
}

```

```

        startActivity(intent);

    }

    } catch (JSONException e) {
        e.printStackTrace();
    }
}
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        new AlertDialog.Builder(SplashScreenActivity.this)
            .setTitle("Oops, Terjadi kesalahan di server.")
            .setMessage("Coba beberapa saat lagi.")
            .setCancelable(false)
            .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface
dialogInterface, int i) {
                    finishAffinity();
                    System.exit(0);
                }
            }).show();
    }
});
MySingleton.getInstance(getApplicationContext()).addToRequestQueue(request)
;

}
}

```

• Login & SignUp Activity

```

package com.rku.meli;

import android.app.AlertDialog;
import android.app.Dialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.os.AsyncTask;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.design.widget.TextInputLayout;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.Adapter;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

```

```

import com.android.volley.DefaultRetryPolicy;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.muddzdev.styleabletoastlibrary.StyleableToast;
import com.rku.meli.SQLite.Session;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.ArrayList;

import javax.net.ssl.HttpURLConnection;

/**
 * Created by Korio Utoro on 9/27/2018.
 */

public class LoginActivity extends AppCompatActivity {

    private Session session;

    TextInputLayout ti_login_username, ti_login_password;
    TextInputLayout ti_signup_password, ti_signup_namadepan,
    ti_signup_namabelakang, ti_signup_email, ti_signup_nrp;
    String login_username, login_password;
    String signup_namadepan, signup_namabelakang, signup_email,
    signup_password, signup_nrp;

    String response_id, response_nama, response_foto, response_stats,
    user_id;

    Button tbl_login, tbl_daftar;
    TextView tvLupaPassword, tvDaftar, tvKembaliLogin;
    LinearLayout formLogin, formSignUp;

    ProgressDialog progressDialog;

    Spinner spinJurusan;
    String jsonJurusan = "https://meli.voidsolution.com/api/jurusan_api/";

    // back button pressed
    private long backPressedTime;
    private Toast backToast;

    int isiMajor;

    @Override
    public void onBackPressed() {
        if (backPressedTime + 2000 > System.currentTimeMillis())
        {
            backToast.cancel();
            konfirmasiKeluar();
        }
        else

```

```

        {
            backToast = Toast.makeText(getApplicationContext(), "Tekan
tombol back sekali lagi untuk keluar", Toast.LENGTH_LONG);
            backToast.show();
        }

        backPressedTime = System.currentTimeMillis();
    }

    private void konfirmasiKeluar()
    {
        new AlertDialog.Builder(LoginActivity.this)
            .setTitle("Anda Yakin Mau Keluar?")
            .setMessage("Anda akan keluar dari aplikasi MeLI")
            .setCancelable(false)
            .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface,
int i) {
                    finishAffinity();
                    System.exit(0);
                }
            })
            .setNegativeButton("BATAL", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface,
int i) {
                    dialogInterface.cancel();
                }
            })
            .show();
    }

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

        session = new Session(this);

        ti_login_username = (TextInputLayout)
findViewById(R.id.login_username);
        ti_login_password = (TextInputLayout)
findViewById(R.id.login_password);

        ti_signup_namadepan = (TextInputLayout)
findViewById(R.id.signup_namadepan);
        ti_signup_namabelakang = (TextInputLayout)
findViewById(R.id.signup_namabelakang);
        ti_signup_email = (TextInputLayout)
findViewById(R.id.signup_email);
        ti_signup_password = (TextInputLayout)
findViewById(R.id.signup_password);
        ti_signup_nrp = (TextInputLayout) findViewById(R.id.signup_nrp);

        tbl_login = (Button) findViewById(R.id.btn_login);
        tbl_daftar = (Button) findViewById(R.id.btn_signup);
        tvLupaPassword = (TextView) findViewById(R.id.btn_lupa_password);
        tvDaftar = (TextView) findViewById(R.id.btn_register);
        tvKembaliLogin = (TextView) findViewById(R.id.btn_kembali_login);
    }

```

```

formLogin = (LinearLayout) findViewById(R.id.form_login);
formSignUp = (LinearLayout) findViewById(R.id.form_signup);

spinJurusan = (Spinner) findViewById(R.id.signup_jurusan);

isiMajor = 0;

progressDialog = new ProgressDialog(LoginActivity.this);

formSignUp.setVisibility(View.GONE);

tvLupaPassword.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        final Dialog cDialog = new Dialog(LoginActivity.this);
        final TextInputLayout ti_email;
        Button btn_batal, btn_kirim;

        cDialog.setContentview(R.layout.dialog_lupa_password);

        ti_email = (TextInputLayout)
cDialog.findViewById(R.id.dialog_email);
        btn_batal = (Button)
cDialog.findViewById(R.id.dialog_btn_cancel);
        btn_kirim = (Button)
cDialog.findViewById(R.id.dialog_btn_kirim);

        cDialog.setCanceledOnTouchOutside(false);
        cDialog.getWindow().setBackgroundDrawable(new
ColorDrawable(Color.TRANSPARENT));
        cDialog.show();

        btn_batal.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                cDialog.dismiss();
            }
        });

        btn_kirim.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                TextInputLayout ti_pass1, ti_pass2;
                Button btn_update;

cekEmail(ti_email.getEditText().getText().toString());
                cDialog.dismiss();
                progressDialog.setTitle("Mengirim Permintaan");
                progressDialog.setMessage("Harap tunggu, sedang
proses permintaan...");
                progressDialog.setCancelable(false);
                progressDialog.show();

            }
        });
    }
});

```

```

tbl_login.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        login_username =
ti_login_username.getText().getText().toString();
        login_password =
ti_login_password.getText().getText().toString();
        doLogin(login_username, login_password);
    }
});

tvDaftar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        formLogin.setVisibility(View.GONE);
        resetFormSignUp();
        formSignUp.setVisibility(View.VISIBLE);
        new Downloader(LoginActivity.this, jsonJurusan,
spinJurusan).execute();
    }
});

tvKembaliLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        resetFormLogin();
        formLogin.setVisibility(View.VISIBLE);
        formSignUp.setVisibility(View.GONE);
    }
});

tbl_daftar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        signup_namadepan =
ti_signup_namadepan.getText().getText().toString();
        signup_namabelakang =
ti_signup_namabelakang.getText().getText().toString();
        signup_email =
ti_signup_email.getText().getText().toString();
        signup_password =
ti_signup_password.getText().getText().toString();
        signup_nrp =
ti_signup_nrp.getText().getText().toString();
        doSignUp(signup_namadepan, signup_namabelakang,
signup_email, signup_password, signup_nrp, isiMajor);
    }
});

}

private void resetFormLogin()
{
    ti_login_username.getText().setText("");
    ti_login_password.getText().setText("");
}

private void resetFormSignUp()
{
    ti_signup_namadepan.getText().setText("");
    ti_signup_namabelakang.getText().setText("");
}

```

```

        ti_signup_email.getEditText().setText("");
        ti_signup_password.getEditText().setText("");
        ti_signup_nrp.getEditText().setText("");
    }

    private void cekEmail(String email)
    {
        String url_cek_email =
"https://meli.voidsolution.com/api/cek_email_api/";

        Log.d("KIRIM EMAIL", email);

        JSONObject params = new JSONObject();
        try {
            params.put("email", email);
        } catch (JSONException e) {
            e.printStackTrace();
        }

        JsonObjectRequest requestCekEmail = new
JsonObjectRequest(url_cek_email, params, new Response.Listener<JSONObject>()
{
    @Override
    public void onResponse(JSONObject response) {

        try {
            int stats = response.getInt("status");

            Log.d("RESPONSE CEK EMAIL", response.toString());

            progressDialog.cancel();

            if (stats == 0)
            {
                StyleableToast.makeText(LoginActivity.this, "Email
Tidak Terdaftar", R.style.warningToast).show();
            }else
            {

                user_id = response.getString("user_id");

                Intent fIntent = new Intent(LoginActivity.this,
ForgetPasswordActivity.class);
                fIntent.putExtra("ID", user_id);

                progressDialog.setTitle("Mengirim Permintaan");
                progressDialog.setMessage("Harap tunggu, sedang
proses permintaan...");
                progressDialog.setCancelable(false);
                progressDialog.show();

                startActivity(fIntent);

            }

        } catch (JSONException e) {
            e.printStackTrace();
        }

    }

}, new Response.ErrorListener() {

```

```

        @Override
        public void onErrorResponse(VolleyError error) {

        }

    });

    requestCekEmail.setRetryPolicy(new DefaultRetryPolicy(
        0,
        DefaultRetryPolicy.DEFAULT_MAX_RETRIES,
        DefaultRetryPolicy.DEFAULT_BACKOFF_MULT
    ));

    MySingleton.getInstance(getApplicationContext()).addToRequestQueue(requestCekEmail);

    }

    private void openPassBaruDialog(String user_id)
    {
        final Dialog pDialog = new Dialog(LoginActivity.this);
        final TextInputLayout ti_pass1, ti_pass2;
        final String id_user = user_id;
        Button btn_update;

        pDialog.setContentView(R.layout.dialog_lupa_password);

        ti_pass1 = (TextInputLayout)
pDialog.findViewById(R.id.dialog_password1);
        ti_pass2 = (TextInputLayout)
pDialog.findViewById(R.id.dialog_password2);
        btn_update = (Button) pDialog.findViewById(R.id.dialog_btn_update);

        pDialog.setCanceledOnTouchOutside(false);
        pDialog.getWindow().setBackgroundDrawable(new
ColorDrawable(Color.TRANSPARENT));
        pDialog.show();

        btn_update.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                if
(ti_pass1.getText().getText().toString().equals(ti_pass2.getText().
getText().toString()))
                {
                    pDialog.dismiss();
                    progressDialog.setTitle("Mengirim Permintaan");
                    progressDialog.setMessage("Harap tunggu, sedang proses
permintaan...");
                    progressDialog.setCancelable(false);
                    progressDialog.show();
                }else
                {
                    StyleableToast.makeText(LoginActivity.this, "Pengisian
Password Berbeda", R.style.warningToast).show();
                    ti_pass1.getText().requestFocus();
                }
            }
        });
    }
}

```

```

private void doLogin(final String username, final String password)
{
    String url_login = "https://meli.voidsolution.com/api/login_api/";

    Log.d("INPUT USERNAME", username);

    JSONObject params = new JSONObject();
    try {
        params.put("uname", username);
        params.put("pass", password);
    } catch (JSONException e) {
        e.printStackTrace();
    }

    JsonObjectRequest requestLogin = new JsonObjectRequest(url_login,
params, new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {
            Log.d("STATUS LOGIN", response.toString());
            int status_login;

            try
            {
                status_login = response.getInt("status");

                if (status_login == 0)
                {
                    new AlertDialog.Builder(LoginActivity.this)
                        .setTitle("Oops")
                        .setMessage("Cek kembali username dan
password Anda!")
                        .setCancelable(false)
                        .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface
dialogInterface, int i) {
                                dialogInterface.cancel();
                            }
                        }).show();
                }
            }
            else
            {
                JSONObject dataResponse =
response.getJSONObject("data");
                response_id = dataResponse.getString("id");
                response_nama = dataResponse.getString("nama");
                response_foto = dataResponse.getString("url_foto");
                response_stats = dataResponse.getString("status");

                progressDialog.setTitle("Selamat Datang, " +
response_nama);
                progressDialog.setMessage("Harap tunggu, sedang
proses login...");
                progressDialog.setCancelable(false);
                progressDialog.show();

                session.setIdLogin(response_id);
                session.setLoggedIn(true);

                Intent intent = new Intent(LoginActivity.this,

```

```

DashboardActivity.class);
        intent.putExtra("ID", response_id);
        intent.putExtra("NAMA", response_nama);
        intent.putExtra("FOTO", response_foto);
        intent.putExtra("STATUS", response_stats);
        startActivity(intent);
    }
} catch (JSONException e) {
    e.printStackTrace();
}

}
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        Log.d("ERROR LOGIN", error.toString());
        if
(error.toString().equals("com.android.volley.ClientError"))
        {
            StyleableToast.makeText(LoginActivity.this, "Login
Gagal", R.style.warningToast).show();
        }
    }
});

MySingleton.getInstance(getApplicationContext()).addToRequestQueue(requestL
ogin);

}

private void doSignUp(final String nama_depan, final String
nama_belakang, final String email, final String password, final String nrp,
final int jurusan_id)
{
    String url_signup =
"https://meli.voidsolution.com/api/signup_api/";

    JSONObject params = new JSONObject();
    try {
        params.put("namadepan", nama_depan);
        params.put("namabelakang", nama_belakang);
        params.put("email", email);
        params.put("password", password);
        params.put("nrp", nrp);
        params.put("jurusan_id", jurusan_id);

        progressDialog.setTitle("Proses Pendaftaran Baru");
        progressDialog.setMessage("Harap tunggu...");
        progressDialog.setCancelable(false);
        progressDialog.show();

    } catch (JSONException e) {
        e.printStackTrace();
    }

    JsonObjectRequest requestSignUp = new JsonObjectRequest(url_signup,
params, new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {
            progressDialog.dismiss();

```

```

        Log.d("RESPONSE SIGNUP", response.toString());
        int status_signup;
        String msg_signup;

        try {
            status_signup = response.getInt("status");

            if (status_signup == 0)
            {
                msg_signup = response.getString("msg");
                new AlertDialog.Builder(LoginActivity.this)
                    .setTitle("Oops")
                    .setMessage(msg_signup)
                    .setCancelable(false)
                    .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface
dialogInterface, int i) {
                            dialogInterface.cancel();
                        }
                    }).show();
            }
            else
            {
                ti_login_username.getText().setText(email);
                ti_login_password.getText().setText(password);

                StyleableToast.makeText(LoginActivity.this,
"Pendaftaran Berhasil", R.style.newEntryToast).show();

                formLogin.setVisibility(View.VISIBLE);
                formSignUp.setVisibility(View.GONE);

            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {

    }
});

requestSignUp.setRetryPolicy(new DefaultRetryPolicy(
    0,
    DefaultRetryPolicy.DEFAULT_MAX_RETRIES,
    DefaultRetryPolicy.DEFAULT_BACKOFF_MULT
));

MySingleton.getInstance(getApplicationContext()).addToRequestQueue(requestS
ignUp);

}

class Downloader extends AsyncTask<Void, Void, String>
{
    Context ctx;

```

```

String jsonUrl, forParser;
Spinner spin;

public Downloader(Context ctx, String jsonUrl, Spinner spin) {
    this.ctx = ctx;
    this.jsonUrl = jsonUrl;
    this.spin = spin;
}

@Override
protected String doInBackground(Void... voids) {
    return this.download();
}

@Override
protected void onPostExecute(String jsonData) {
    super.onPostExecute(jsonData);

    spin.setAdapter(null);

    if (jsonData.startsWith("Error"))
    {
        String error = jsonData;
        Toast.makeText(ctx, error, Toast.LENGTH_SHORT).show();
    }else
    {
        new JSONParserMajor(ctx, jsonData, spin).execute();
    }
}

private String download()
{
    Object connection = Connector.connect(jsonUrl);
    if (connection.toString().startsWith("Error"))
    {
        return connection.toString();
    }

    try
    {
        HttpURLConnection conn = (HttpURLConnection) connection;
        if (conn.getResponseCode() == conn.HTTP_OK)
        {
            InputStream is = new
BufferedInputStream(conn.getInputStream());
            BufferedReader br = new BufferedReader(new
InputStreamReader(is));

            String line;
            StringBuffer jsonData = new StringBuffer();

            //READ DATA

            while ((line = br.readLine()) != null)
            {
                jsonData.append(line+"\n");
            }

            // CLOSE RESOURCES
            br.close();
            is.close();

```

```

        // RETURN JSON
        return jsonData.toString();

    }else
    {
        return "Error "+conn.getResponseMessage();
    }
} catch (IOException e) {
    e.printStackTrace();
    return "Error "+e.getMessage();
}
}

private class JSONParserMajor extends AsyncTask<Void, Void, Boolean> {

    Context ctx;
    String jsonData;
    int idMajor;
    Spinner spin;

    public JSONParserMajor(Context ctx, String jsonData, Spinner spin)
    {
        this.ctx = ctx;
        this.jsonData = jsonData;
        this.spin = spin;
    }

    ArrayList<String> majorList = new ArrayList<>();
    ArrayList<Integer> idList = new ArrayList<>();

    @Override
    protected Boolean doInBackground(Void... voids) {
        return this.parse();
    }

    @Override
    protected void onPostExecute(Boolean isParsed) {
        super.onPostExecute(isParsed);

        if (isParsed)
        {
            ArrayAdapter<String> adapter = new
ArrayAdapter<String>(ctx, android.R.layout.simple_spinner_dropdown_item,
majorList)
            {
                @NonNull
                @Override
                public View getView(int position, @Nullable View
convertView, @NonNull ViewGroup parent) {
                    View view = super.getView(position, convertView,
parent);

                    if(position == getCount())
                    {
                        ((TextView)
view.findViewById(android.R.id.text1)).setText("");
                        ((TextView)
view.findViewById(android.R.id.text1)).setHint(getItem(getCount()));

```

```

        }

        return view;
    }

    @Override
    public int getCount() {
        return super.getCount() - 1;
    }
};

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

spin.setAdapter(adapter);
spin.setSelection(adapter.getCount());

spin.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> parent, View
view, int position, long id) {
        isiMajor = idList.get(position);
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {
    }

});

}else
{
    Toast.makeText(ctx, "Gagal Memuat Jurusan",
Toast.LENGTH_SHORT).show();
}

}

private Boolean parse()
{
    try
    {
        JSONObject root = new JSONObject(jsonData);
        JSONArray ja = root.getJSONArray("data");
        JSONObject jo;

        majorList.clear();
        for (int i = 0; i < ja.length(); i++)
        {
            jo = ja.getJSONObject(i);

            int idMajorList = jo.getInt("id");
            String nameMajorList = jo.getString("name");
            idMajor = idMajorList; // it start with 0 index
            idList.add(idMajor);
            majorList.add(nameMajorList);
        }
        majorList.add("-- Pilih Jurusan --");
        idList.add(0);
    }
}

```

```

        return true;

    } catch (JSONException e) {
        e.printStackTrace();
        return false;
    }
}
}
}
}

```

- **Dashboard Activity**

```

package com.rku.meli;

import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.support.design.widget.TabLayout;
import android.support.v4.view.ViewPager;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.muddzdev.styleabletoastlibrary.StyleableToast;
import com.rku.meli.SQLite.Session;
import com.squareup.picasso.Picasso;

import org.json.JSONException;
import org.json.JSONObject;

/**
 * Created by Korio Utoro on 9/27/2018.
 */

public class DashboardActivity extends AppCompatActivity {

    private Session session;

    private TabLayout tabLayout;
    private ViewPager viewPager;
    private ProgressBar pgFoto;

    TextView tv_profil_nama;
    ViewPagerMenuCourseAdapter adapter;
    LinearLayout llWarning, llTab;
    ImageView iv_oops, profile_foto;

```

```

    Uri fotona;

    Button btn_notif, btn_setting, btn_refresh;

    String user_id, nama_depan, nama_belakang, email, phone1, institution,
    address, city, country, url_foto, status;
    String display_name;

    ProgressDialog progressDialog;

    ProgressBar progressBar;

    // back button pressed
    private long backPressedTime;
    private Toast backToast;

    @Override
    public void onBackPressed() {
        if (backPressedTime + 2000 > System.currentTimeMillis())
        {
            backToast.cancel();
            finishAffinity();
            System.exit(0);
        }else
        {
            backToast = Toast.makeText(getApplicationContext(), "Tekan
tombol back sekali lagi untuk keluar", Toast.LENGTH_LONG);
            backToast.show();
        }

        backPressedTime = System.currentTimeMillis();
    }

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

        session = new Session(this);

        progressDialog = new ProgressDialog(DashboardActivity.this);
        progressBar = (ProgressBar) findViewById(R.id.dashboard_progress);
        pgFoto = (ProgressBar) findViewById(R.id.dashboard_pg_foto);

        tv_profil_nama = (TextView)
        findViewById(R.id.dashboard_profil_nama);
        llWarning = (LinearLayout)
        findViewById(R.id.dashboard_layout_warning);
        llTab = (LinearLayout) findViewById(R.id.dashboard_layout_tab);
        iv_oops = (ImageView) findViewById(R.id.dashboard_image_oops);

        profile_foto = (ImageView)
        findViewById(R.id.dashboard_profil_foto);

        btn_setting = (Button) findViewById(R.id.btn_dashboard_setting);
        btn_refresh = (Button) findViewById(R.id.dashboard_btn_refresh);

        tabLayout = (TabLayout) findViewById(R.id.dashboard_tabs);
        viewPager = (ViewPager) findViewById(R.id.dashboard_vp);

```

```

String stats = session.getStatusUser();
Log.d("STATUS SESSION", stats);

user_id = session.getIdLogin();
int userID = Integer.valueOf(user_id);

if (stats.equals("1"))
{
    initUser(userID);
    String nama_user = session.getUserName();
    tv_profil_nama.setText(nama_user);
    progressBar.setVisibility(View.GONE);
    llWarning.setVisibility(View.GONE);
    llTab.setVisibility(View.VISIBLE);
}else
{
    initUser(userID);
}

btn_refresh.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        progressDialog.setTitle("Proses Pembaruan");
        progressDialog.setMessage("Harap tunggu, sedang proses
pemburuan data...");
        progressDialog.setCancelable(false);
        progressDialog.show();

        int id_user = Integer.valueOf(user_id);
        initUser(id_user);
    }
});

btn_setting.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new Intent(DashboardActivity.this,
SettingActivity.class));
    }
});

adapter = new
ViewPagerMenuCourseAdapter(getSupportFragmentManager());

adapter.AddFragment(new MenuMyCourseFragment(),"Kelas Saya");
adapter.AddFragment(new MenuListCourseFragment(),"List Seluruh
Kelas");

viewPager.setAdapter(adapter);
tabLayout.setupWithViewPager(viewPager);
}

private void initUser(final int userID)
{
    String url_get_user =
"https://meli.voidsolution.com/api/get_user_api/";

```

```

        JsonObjectRequest request = new
        JsonObjectRequest(Request.Method.GET, url_get_user + "?user_id=" + userID,
        null, new Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject response) {

                Log.d("RESPONSE GET USER", response.toString());

                try {
                    int status_getUser = response.getInt("status");

                    if (status_getUser == 0)
                    {

                        String msg_getUser = response.getString("msg");

                        new AlertDialog.Builder(DashboardActivity.this)
                            .setTitle("Oops")
                            .setMessage(msg_getUser)
                            .setCancelable(false)
                            .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface
dialogInterface, int i) {

                                    dialogInterface.cancel();

                                }
                            }).show();

                    }else
                    {

                        progressDialog.cancel();

                        JSONObject dataResponse =
response.getJSONObject("data");

                        user_id =
String.valueOf(dataResponse.getInt("id"));
                        if
(dataResponse.getString("namadepan").equals("null"))
                        {
                            nama_depan = "";
                        }else
                        {
                            nama_depan =
dataResponse.getString("namadepan");
                        }
                        if
(dataResponse.getString("namabelakang").equals("null"))
                        {
                            nama_belakang = "";
                        }else
                        {
                            nama_belakang =
dataResponse.getString("namabelakang");
                        }
                        if (dataResponse.getString("email").equals("null"))
                        {
                            email = "";
                        }else

```

```

        {
            email = dataResponse.getString("email");
        }
        if (dataResponse.getString("phone").equals("null"))
        {
            phone1 = "";
        }else
        {
            phone1 = dataResponse.getString("phone");
        }
        phone1 = dataResponse.getString("phone");

        if
(dataResponse.getString("address").equals("null"))
        {
            address = "";
        }else
        {
            address = dataResponse.getString("address");
        }
        if (dataResponse.getString("city").equals("null"))
        {
            city = "";
        }else
        {
            city = dataResponse.getString("city");
        }
        if
(dataResponse.getString("country").equals("null"))
        {
            country = "";
        }else
        {
            country = dataResponse.getString("country");
        }
        if
(dataResponse.getString("url_foto").equals("null"))
        {
            url_foto = "";
        }else
        {
            url_foto = dataResponse.getString("url_foto");
        }
        status = dataResponse.getString("status");

        display_name = nama_depan.toUpperCase() + " " +
nama_belakang.toUpperCase();

        tv_profil_nama.setText(display_name);

        session.setIdLogin(user_id);
        session.setStatusUser(status);
        session.setNamaDepan(nama_depan);
        session.setNamaBelakang(nama_belakang);
        session.setUserName(display_name);
        session.setEmailUser(email);
        session.setPhoneUser(phone1);
        session.setAddressUser(address);
        session.setCityUser(city);
        session.setCountryUser(country);
        session.setFotoUser(url_foto);

```

```

        fotona = Uri.parse(session.getFotoUser());

        if (session.getFotoUser().equals("") ||
session.getFotoUser().equals("null"))
        {
            pgFoto.setVisibility(View.GONE);
            Picasso.get()
                .load(R.drawable.ic_account)
                .placeholder(R.drawable.ic_account)
                .error(R.drawable.ic_account)
                .into(profile_foto);
        }else
        {
            pgFoto.setVisibility(View.GONE);
            Picasso.get()
                .load(fotona.toString())
                .placeholder(R.drawable.ic_account)
                .error(R.drawable.ic_account)
                .into(profile_foto);
        }

        Log.d("USER ID SEKARANG", user_id);

        if (status.equals("1"))
        {
            StyleableToast.makeText(DashboardActivity.this,
"Pembaruan Data Selesai", R.style.updatedToast).show();
            progressBar.setVisibility(View.GONE);
            llWarning.setVisibility(View.GONE);
            llTab.setVisibility(View.VISIBLE);
        }else
        {
            progressBar.setVisibility(View.GONE);
            llWarning.setVisibility(View.VISIBLE);
            llTab.setVisibility(View.GONE);
        }

    }

    } catch (JSONException e) {
        e.printStackTrace();
    }
}

}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {

    }

});

MySingleton.getInstance(getApplicationContext()).addToRequestQueue(request)
;

    }

}

```

- **Detail Course Activity**

```
package com.rku.meli;

import android.annotation.SuppressLint;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.support.design.widget.TabLayout;
import android.support.v4.view.ViewPager;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Html;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;

import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.rku.meli.SQLite.Session;

import org.json.JSONException;
import org.json.JSONObject;

/**
 * Created by Korio Utoro on 9/27/2018.
 */

public class DetailCourseActivity extends AppCompatActivity {

    private TabLayout tabLayout;
    private ViewPager viewPager;

    Session session;
    ProgressDialog progressDialog;
    TextView nama_course, deskripsi_course;
    String status_course, id_course, id_enroll;
    String id_user, display_nama;
    LinearLayout llOverview, llTab;
    ViewPagerDetailCourseSectionAdapter adapter;
    Button btn_back, btn_aksi;

    @SuppressWarnings("NewApi", "ResourceAsColor")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail_course);

        session = new Session(DetailCourseActivity.this);
        id_user = session.getIdLogin();

        progressDialog = new ProgressDialog(DetailCourseActivity.this);

        tabLayout = (TabLayout) findViewById(R.id.detail_course_tabs);
        viewPager = (ViewPager) findViewById(R.id.detail_course_vp);

        nama_course = (TextView) findViewById(R.id.detail_course_tv_nama);
        deskripsi_course = (TextView)
```

```

findViewById(R.id.detail_course_tv_overview);

    llOverview = (LinearLayout)
findViewById(R.id.detail_course_layout_overview);
    llTab = (LinearLayout) findViewById(R.id.detail_course_layout_tab);

    btn_back = (Button) findViewById(R.id.detail_course_btn_back);
    btn_aksi = (Button) findViewById(R.id.detail_course_btn_aksi);

    adapter = new
ViewPagerDetailCourseSectionAdapter(getSupportFragmentManager());

    adapter.AddFragment(new DetailCourseOverviewFragment(), "IKHTISAR");
    adapter.AddFragment(new DetailCourseSectionFragment(), "SESI");

    viewPager.setAdapter(adapter);
    tabLayout.setupWithViewPager(viewPager);

    status_course = session.getStatusEnroll();
    display_nama = session.getJudulCourse();
    id_course = session.getIdCourse();
    id_enroll = session.getIdEnroll();

    Log.d("SESSION IDCOURSE", id_course);
    Log.d("SESSION IDENROLL", id_enroll);

    nama_course.setText(display_nama.toUpperCase());

    if (status_course.equals("1"))
    {
        llOverview.setVisibility(View.GONE);
        llTab.setVisibility(View.VISIBLE);
        btn_aksi.setVisibility(View.GONE);
    }else
    {
        llOverview.setVisibility(View.VISIBLE);
        llTab.setVisibility(View.GONE);
        btn_aksi.setVisibility(View.VISIBLE);
    }

    btn_aksi.setBackgroundTintList(getResources().getColorStateList(R.color.col
orBtnEnroll));

    deskripsi_course.setText(Html.fromHtml(session.getDeskripsiCourse()));
    btn_aksi.setText("ENROLL");
    btn_aksi.setTextColor(R.color.colorPrimary);
}

    btn_aksi.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            requestPassEnroll(Integer.valueOf(id_user),
Integer.valueOf(id_course), Integer.valueOf(id_enroll));
            progressDialog.setTitle("Mengirimkan Permintaan");
            progressDialog.setMessage("Harap tunggu, sedang proses
enrollment...");
            progressDialog.setCancelable(false);
            progressDialog.show();
        }
    });
}

```

```

        btn_back.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(DetailCourseActivity.this,
DashboardActivity.class));
            }
        });

    }

    private void requestPassEnroll(final int id_user, int id_course, final
int id_enroll)
    {
        String url_api =
"https://meli.voidsolution.com/api/minta_pass_api/";

        JSONObject params = new JSONObject();
        try {
            params.put("user_id", id_user);
            params.put("course_id", id_course);
            params.put("enroll_id", id_enroll);
        } catch (JSONException e) {
            e.printStackTrace();
        }

        final JsonObjectRequest requestPass = new
JsonObjectRequest(url_api, params, new Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject response) {

                Log.d("RESPONSE MINTA PASS", response.toString());
                int status_minta;

                try {
                    status_minta = response.getInt("status");

                    if (status_minta == 0)
                    {
                        progressDialog.cancel();
                        new AlertDialog.Builder(DetailCourseActivity.this)
                            .setTitle("Oops")
                            .setMessage("Terjadi kesalahan sistem.
Silakan coba beberapa saat lagi!")
                            .setCancelable(false)
                            .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface
dialogInterface, int i) {
                                    dialogInterface.cancel();
                                }
                            }).show();
                    }else
                    {
                        String email = response.getString("email");
                        Intent mIntent = new
Intent(DetailCourseActivity.this, EnrollPasswordActivity.class);

                        session.setEmailUser(email);
                        session.setEnrollID(String.valueOf(id_enroll));
                        session.setIdLogin(String.valueOf(id_user));
                    }
                }
            }
        });
    }

```

```

        progressDialog.cancel();
        startActivity(mIntent);
    }

    } catch (JSONException e) {
        e.printStackTrace();
    }

    }

}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {

    }

});

MySingleton.getInstance(getApplicationContext()).addToRequestQueue(requestP
ass);
    }
}

```

• Detail Course Overview Fragment

```

package com.rku.meli;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.widget.SwipeRefreshLayout;
import android.text.Html;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ProgressBar;
import android.widget.TextView;

import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.muddzdev.styleabletoastlibrary.StyleableToast;
import com.rku.meli.SQLite.Session;

import org.json.JSONException;
import org.json.JSONObject;

/**
 * A simple {@link Fragment} subclass.
 */

/**
 * Created by Korio Utoro on 9/27/2018.
 */

public class DetailCourseOverviewFragment extends Fragment {

    private ProgressBar progressBar;
    private SwipeRefreshLayout refreshLayout;

    Session session;

```

```

        TextView tv_deskripsi;

        int id_course;

        public DetailCourseOverviewFragment() {
            // Required empty public constructor
        }

        @Override
        public View onCreateView(LayoutInflater inflater, ViewGroup container,
                                Bundle savedInstanceState) {

            View rootView =
inflater.inflate(R.layout.fragment_detail_course_overview, container,
false);

            session = new Session(getContext());
            id_course = Integer.valueOf(session.getIdCourse());

            tv_deskripsi = (TextView)
rootView.findViewById(R.id.overview_deskripsi);

            progressBar = (ProgressBar)
rootView.findViewById(R.id.overview_progressbar);
            refreshLayout = (SwipeRefreshLayout)
rootView.findViewById(R.id.sw_overview);

            getDetailCourse(id_course);

            refreshLayout.setOnRefreshListener(new
SwipeRefreshLayout.OnRefreshListener() {
                @Override
                public void onRefresh() {
                    getDetailCourse(id_course);
                }
            });

            return rootView;
        }

        private void getDetailCourse(int id_course)
        {
            String url_get_detail_course
="https://meli.voidsolution.com/api/course_detail_api/";

            JSONObjectRequest requestDetail = new
JsonObjectRequest(JsonObjectRequest.Method.GET, url_get_detail_course +
"?course_id=" + id_course, null, new Response.Listener<JSONObject>() {
                @Override
                public void onResponse(JSONObject response) {

                    Log.d("RESPONSE DETAIL", response.toString());

                    try {
                        int status_response = response.getInt("status");

                        refreshLayout.setRefreshing(false);
                        progressBar.setVisibility(View.GONE);

                        if (status_response == 0)

```

```

        {
            StyleableToast.makeText(getContext(), "Terjadi
kesalahan, swipe ke bawah untuk refresh!", R.style.warningToast).show();
        } else
        {
            JSONObject dataResponse =
response.getJSONObject("data");
tv_deskripsi.setText(Html.fromHtml(dataResponse.getString("deskripsi")));
        }

        } catch (JSONException e) {
            e.printStackTrace();
        }

    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {

    }
});

MySingleton.getInstance(getContext()).addToRequestQueue(requestDetail);
}
}

```

- Detail Course Section Fragment

```

package com.rku.meli;

import android.content.Intent;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.widget.SwipeRefreshLayout;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ExpandableListView;
import android.widget.ProgressBar;

import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.muddzdev.styleabletoastlibrary.StyleableToast;
import com.rku.meli.SQLite.Session;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.LinkedHashMap;

/**
 * A simple {@link Fragment} subclass.
 */

```

```

* Created by Korio Utoro on 9/27/2018.
*/

public class DetailCourseSectionFragment extends Fragment {

    private SwipeRefreshLayout refreshLayout;
    private ProgressBar progressBar;

    Session session;
    int id_course;

    private ExpandableListView listView;

    private LinkedHashMap<String, ModelSection> subjects = new
LinkedHashMap<String, ModelSection>();
    private ArrayList<ModelSection> section = new
ArrayList<ModelSection>();
    private ArrayList<ModelSubSection> subSection;

    private CourseSectionAdapter sectionAdapter;

    public DetailCourseSectionFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {

        View rootView =
inflater.inflate(R.layout.fragment_detail_course_section, container,
false);

        session = new Session(getContext());
        id_course = Integer.valueOf(session.getIdCourse());

        progressBar = (ProgressBar)
rootView.findViewById(R.id.section_progressbar);

        listView = (ExpandableListView)
rootView.findViewById(R.id.lv_section);

        initData();

        listView.setOnChildClickListener(new
ExpandableListView.OnChildClickListener() {
            @Override
            public boolean onChildClick(ExpandableListView parent, View
view, int groupPosition, int childPosition, long id) {

                // get group header
                ModelSection modelSection = section.get(groupPosition);

                // get subsection
                ModelSubSection modelSubSection =
modelSection.getSubSection().get(childPosition);

                String deskripsi_sec =
modelSubSection.getDeskripsiSubSection();
                String kode_embed = modelSubSection.getKode_embed();

```

```

        String file_pdf = modelSubSection.getFile_url();
        String link = modelSubSection.getLink();
        String quiz_id = modelSubSection.getQuiz_id();
        String tugas_id = modelSubSection.getTugas_id();
        String judul = modelSubSection.getJudul();

        if (judul.toLowerCase().equals("tidak ada subsection"))
        {
            StyleableToast.makeText(getContext(), "TIDAK DAPAT
MENAMPILKAN ISI", R.style.warningToast).show();
        }else {

            Intent vIntent = new Intent(getActivity(),
DetailVideoActivity.class);

            session.setDeskripsiVideo(deskripsi_sec);
            session.setKodeVideo(kode_embed);
            session.setPdfVideo(file_pdf);
            session.setLinkTerkait(link);
            session.setIdQuiz(quiz_id);
            session.setIdTugas(tugas_id);

            startActivity(vIntent);
        }

        return false;
    }
});

return rootView;
}

private void initData()
{
    progressBar.setVisibility(View.GONE);

    getDetailCourse(id_course);
}

private void getDetailCourse(int id_course)
{
    String url_get_detail_course
="https://meli.voidsolution.com/api/course_detail_api/";

    JSONObjectRequest requestDetail = new
JSONObjectRequest(JSONObjectRequest.Method.GET, url_get_detail_course +
"?course_id=" + id_course, null, new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {

            Log.d("RESPONSE DETAIL", response.toString());

            try {
                int status_response = response.getInt("status");

                progressBar.setVisibility(View.GONE);

                if (status_response == 0)
                {

```

```

        StyleableToast.makeText(getContext(), "Terjadi
kesalahan, swipe ke bawah untuk refresh!", R.style.warningToast).show();
    } else
    {
        JSONObject dataResponse =
response.getJSONObject("data");
        JSONArray jsonArray =
dataResponse.getJSONArray("section");
        Log.d("BANYAK DATA SECTION",
String.valueOf(jsonArray.length()));

        for (int i=0; i<jsonArray.length(); i++)
        {
            JSONObject jsonSubSection =
jsonArray.getJSONObject(i);

            String judul_section =
jsonSubSection.getString("judul");
            String deskripsi_section =
jsonSubSection.getString("deskripsi");

            Log.d("JUDUL SECTION", judul_section);

            JSONArray jsonArraySub =
jsonSubSection.getJSONArray("subsection");
            if (jsonArraySub.length() == 0)
            {
                Log.d("JUDUL SUBSECTION", "Tidak ada
subsection");

                addSection(judul_section,
deskripsi_section, "Tidak ada subsection", "", "", "", "", "");
            }

            for (int j=0; j<jsonArraySub.length(); j++)
            {
                JSONObject jsonIsiSub =
jsonArraySub.getJSONObject(j);
                String judul_sub =
jsonIsiSub.getString("judul");
                String deskripsi_sub =
jsonIsiSub.getString("deskripsi");
                String kode_video =
jsonIsiSub.getString("kode_embed");
                String file_pdf =
jsonIsiSub.getString("file_url");
                String id_quiz =
jsonIsiSub.getString("quiz_id");
                String id_tugas =
jsonIsiSub.getString("tugas_id");

                addSection(judul_section,
deskripsi_section, judul_sub, deskripsi_sub, kode_video, file_pdf, id_quiz,
id_tugas);

                Log.d("JUDUL SUBSECTION", judul_sub);
            }
        }
    }

```

```

        sectionAdapter = new
CourseSectionAdapter(getContext(), section);
        listView.setAdapter(sectionAdapter);

    }

    } catch (JSONException e) {
        e.printStackTrace();
    }

    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {

    }

});

MySingleton.getInstance(getContext()).addToRequestQueue(requestDetail);

}

private int addSection(String judul_section, String deskripsi_section,
String judul_sub, String deskripsi_sub, String kode_video, String file_pdf,
String id_quiz, String id_tugas)
{
    int sectionPosition = 0;

    ModelSection modelSection = subjects.get(judul_section);
    if (modelSection == null)
    {
        modelSection = new ModelSection();
        modelSection.setJudul(judul_section);
        modelSection.setDeskripsi(deskripsi_section);
        subjects.put(judul_section, modelSection);
        section.add(modelSection);
    }

    subSection = modelSection.getSubSection();

    int subSectionSize = subSection.size();

    Log.d("BESAR ISI", String.valueOf(subSectionSize));

    subSectionSize++;

    ModelSubSection detail_subsection = new ModelSubSection();
    detail_subsection.setSequence(String.valueOf(subSectionSize));
    detail_subsection.setJudul(judul_sub);
    detail_subsection.setDeskripsiSubSection(deskripsi_sub);
    detail_subsection.setKode_embed(kode_video);
    detail_subsection.setFile_url(file_pdf);
    detail_subsection.setQuiz_id(id_quiz);
    detail_subsection.setTugas_id(id_tugas);
    subSection.add(detail_subsection);
    sectionPosition = section.indexOf(subSection);

    return sectionPosition;
}
}

```

- **Detail Video Activity**

```
package com.rku.meli;

import android.annotation.SuppressLint;
import android.app.AlertDialog;
import android.app.DownloadManager;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.content.res.Configuration;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.media.AudioManager;
import android.net.Uri;
import android.os.Build;
import android.provider.Settings;
import android.os.Bundle;
import android.text.Html;
import android.util.Log;
import android.view.KeyEvent;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.google.android.youtube.player.YouTubeBaseActivity;
import com.google.android.youtube.player.YouTubeInitializationResult;
import com.google.android.youtube.player.YouTubePlayer;
import com.google.android.youtube.player.YouTubePlayerView;
import com.google.android.youtube.player.YouTubePlayer.ErrorReason;
import com.google.android.youtube.player.YouTubePlayer.PlaybackEventListener;
import com.google.android.youtube.player.YouTubePlayer.PlayerStateChangeListener;
import com.muddzdev.styleabletoastlibrary.StyleableToast;
import com.rku.meli.SQLite.Session;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

/**
 * Created by Korio Utoro on 9/27/2018.
 */

public class DetailVideoActivity extends YouTubeBaseActivity implements
    YouTubePlayer.OnFullscreenListener,
    YouTubePlayer.OnInitializedListener,
    YouTubePlayer.PlayerStateChangeListener {

    public static final String API_KEY =
    "AIzaSyAlyZDAcRuoG1pmDvYK8V661g981OG8HH4";

    Session session;
```

```

String user_id, VIDEO_ID, file_pdf, link_terkait, quiz_id, tugas_id;

String judul_quiz, deskripsi_quiz, durasi_quiz, minimum_nilai_quiz;
String judul_tugas, deskripsi_tugas, deadline_tugas;

JSONArray arrResult;

TextView deskripsi;

Button btn_back, btn_fullscreen, btn_fullscreen_exit, btn_pdf,
btn_link, btn_quiz, btn_tugas;
ImageView logo_atas;
LinearLayout layout_bawah;

DownloadManager downloadManager;

@SuppressLint("InlinedApi")
private static final int PORTRAIT_ORIENTATION = Build.VERSION.SDK_INT <
9
    ? ActivityInfo.SCREEN_ORIENTATION_PORTRAIT
    : ActivityInfo.SCREEN_ORIENTATION_SENSOR_PORTRAIT;

@SuppressLint("InlinedApi")
private static final int LANDSCAPE_ORIENTATION = Build.VERSION.SDK_INT
< 9
    ? ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE
    : ActivityInfo.SCREEN_ORIENTATION_SENSOR_LANDSCAPE;

private YouTubePlayer mPlayer = null;
private boolean mAutoRotation = false;

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

    session = new Session(this);
    user_id = session.getIdLogin();

    /// Inisialisasi YouTube Player View ///
    final YouTubePlayerView youtubePlayerView =
findViewById(R.id.youtube_player);
    youtubePlayerView.initialize(API_KEY, this);

    deskripsi = (TextView) findViewById(R.id.youtube_deskripsi);

    btn_pdf = (Button) findViewById(R.id.youtube_btn_file);
    btn_quiz = (Button) findViewById(R.id.youtube_btn_quiz);

    Log.d("KODE VIDEO", session.getKodeVideo());
    Log.d("DESKRIPSI VIDEO", session.getDeskripsiVideo());
    Log.d("FILE PDF", session.getPdfVideo());

    VIDEO_ID = session.getKodeVideo();
    deskripsi.setText(Html.fromHtml(session.getDeskripsiVideo()));
    file_pdf = session.getPdfVideo();
    if (file_pdf.equals("") || file_pdf.equals("null"))
    {
        btn_pdf.getBackground().setColorFilter(Color.GRAY,
PorterDuff.Mode.MULTIPLY);

```

```

        btn_pdf.setAlpha(.5f);
        btn_pdf.setClickable(false);
    }else
    {
        btn_pdf.setBackground().setColorFilter(null);
        btn_pdf.setClickable(true);
    }

    quiz_id = session.getIdQuiz();
    tugas_id = session.getIdTugas();
    if (quiz_id.equals("") || quiz_id.equals("null"))
    {
        if (tugas_id.equals("") || tugas_id.equals("null"))
        {
            btn_quiz.setBackground().setColorFilter(Color.GRAY,
PorterDuff.Mode.MULTIPLY);
            btn_quiz.setAlpha(.5f);
            btn_quiz.setClickable(false);
        }else
        {
            btn_quiz.setBackground().setColorFilter(null);
            btn_quiz.setClickable(true);
            btn_quiz.setText("ISI TUGAS");
            getTugas(user_id, tugas_id);
        }
    }else
    {
        getQuiz(user_id, quiz_id);
        btn_quiz.setText("IKUTI QUIZ");
        btn_quiz.setBackground().setColorFilter(null);
        btn_quiz.setClickable(true);
    }

    mAutoRotation = Settings.System.getInt(getContentResolver(),
        Settings.System.ACCELEROMETER_ROTATION, 0) == 1;

    logo_atas = (ImageView) findViewById(R.id.youtube_logo_leon_atas);
    btn_fullscreen = (Button)
findViewById(R.id.youtube_btn_fullscreen);
    btn_fullscreen_exit = (Button)
findViewById(R.id.youtube_btn_fullscreen_exit);

    layout_bawah = (LinearLayout)
findViewById(R.id.youtube_layout_bawah);

    btn_fullscreen.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            onFullscreen(true);
            layout_bawah.setVisibility(View.GONE);
            logo_atas.setVisibility(View.VISIBLE);
            btn_fullscreen_exit.setVisibility(View.VISIBLE);
        }
    });

    btn_fullscreen_exit.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            onFullscreen(false);

```

```

        layout_bawah.setVisibility(View.VISIBLE);
        logo_atas.setVisibility(View.GONE);
        btn_fullscreen_exit.setVisibility(View.GONE);
    }
});

btn_pdf.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (file_pdf.equals("") || file_pdf.equals("null"))
        {
            StyleableToast.makeText(DetailVideoActivity.this,
"Dokumen pdf tidak tersedia!", R.style.warningToast).show();
        }
        else
        {
            downloadManager = (DownloadManager)
getSystemService(Context.DOWNLOAD_SERVICE);
            Uri uri = Uri.parse("https://meli.voidsolution.com" +
file_pdf);

            DownloadManager.Request requestPdf = new
DownloadManager.Request(uri);

requestPdf.setNotificationVisibility(DownloadManager.Request.VISIBILITY_VIS
IBLE_NOTIFY_COMPLETED);
            Long reference = downloadManager.enqueue(requestPdf);
            StyleableToast.makeText(DetailVideoActivity.this,
"Dokumen sedang didownload...", R.style.downloadPdfToast).show();
        }
    }
});

btn_quiz.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (quiz_id.equals("") || quiz_id.equals("null"))
        {
            if (tugas_id.equals("") || tugas_id.equals("null"))
            {
                StyleableToast.makeText(DetailVideoActivity.this,
"Quiz tidak tersedia!", R.style.warningToast).show();
            }
            else
            {
                Intent tIntent = new
Intent(DetailVideoActivity.this, TugasActivity.class);
                session.setJudulTugas(judul_tugas);
                session.setDeskripsiTugas(deskripsi_tugas);
                session.setDeadlineTugas(deadline_tugas);
                session.setIdTugas(tugas_id);
                startActivity(tIntent);
            }
        }
        else {
            Intent qIntent = new Intent(DetailVideoActivity.this,
QuizOverviewActivity.class);
            session.setJudulQuiz(judul_quiz);
            session.setDeskripsiQuiz(deskripsi_quiz);
            session.setDurasiQuiz(durasi_quiz);
            session.setMinimumNilaiQuiz(minimum_nilai_quiz);
            session.setIdQuiz(quiz_id);
            session.setArrResult(arrResult.toString());
        }
    }
});

```

```

        session.setBeresQuiz(false);
        Log.d("ARR_RESULT", arrResult.toString());

        startActivity(qIntent);
    }
}

});

btn_back = (Button) findViewById(R.id.btn_detail_back);
btn_back.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view)
    {
        startActivity(new Intent(DetailVideoActivity.this,
DetailCourseActivity.class));
    }
});
}

private void getTugas(String user_id, final String tugas_id)
{
    String url_get_tugas =
"https://meli.voidsolution.com/api/get_tugas_api/";

    JSONObject paramsGetTugas = new JSONObject();
    try {
        paramsGetTugas.put("user_id", Integer.valueOf(user_id));
        paramsGetTugas.put("tugas_id", Integer.valueOf(tugas_id));
    } catch (JSONException e) {
        e.printStackTrace();
    }

    JsonObjectRequest requestGetTugas = new
JsonObjectRequest(url_get_tugas, paramsGetTugas, new
Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {

            try {
                int status_getTugas = response.getInt("status");

                if (status_getTugas == 0)
                {
                    String msg_getTugas = response.getString("msg");

                    new AlertDialog.Builder(DetailVideoActivity.this)
                        .setTitle("Oops")
                        .setMessage(msg_getTugas)
                        .setCancelable(false)
                        .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface
dialogInterface, int i) {

                                dialogInterface.cancel();
                            }
                        }).show();
                }
            } else
            {
                JSONObject tugasInfo =
response.getJSONObject("tugas_info");

```

```

        judul_tugas = tugasInfo.getString("judul");
        deskripsi_tugas =
tugasInfo.getString("description");
        deadline_tugas = tugasInfo.getString("deadline");
        String idTugas =
Integer.valueOf(tugasInfo.getString("tugas_id")).toString();

        session.setIdTugas(idTugas);
        session.setJudulTugas(judul_tugas);
        session.setDeskripsiTugas(deskripsi_tugas);
        session.setDeadlineTugas(deadline_tugas);
    }

    } catch (JSONException e) {
        e.printStackTrace();
    }

    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {

    }
});

MySingleton.getInstance(DetailVideoActivity.this).addToRequestQueue(request
GetTugas);
}

private void getQuiz(String user_id, String quiz_id)
{
    String url_get_quiz =
"https://meli.voidsolution.com/api/get_quiz_api/";

    JSONObject params = new JSONObject();
    try {
        params.put("user_id", Integer.valueOf(user_id));
        params.put("quiz_id", Integer.valueOf(quiz_id));
    } catch (JSONException e) {
        e.printStackTrace();
    }

    JsonObjectRequest requestQuiz = new JsonObjectRequest(url_get_quiz,
params, new Response.Listener<JSONObject>() {
    @Override
    public void onResponse(JSONObject response) {
        try {
            int status_getQuiz = response.getInt("status");

            if (status_getQuiz == 0)
            {
                String msg_getQuiz = response.getString("msg");

                new AlertDialog.Builder(DetailVideoActivity.this)
                    .setTitle("Oops")
                    .setMessage(msg_getQuiz)
                    .setCancelable(false)
                    .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                        @Override

```



```

    }

    if(!wasRestored)
    {
        youtubePlayer.cueVideo(VIDEO_ID);
    }

}

@Override
public void onInitializationFailure(YouTubePlayer.Provider provider,
YouTubeInitializationResult youtubeInitializationResult) {
    Toast.makeText(this, "Gagal melakukan inisialisasi",
Toast.LENGTH_SHORT).show();
}

private PlayerStateChangeListener playerStateChangeListener = new
PlayerStateChangeListener() {
    @Override
    public void onLoading() {

    }

    @Override
    public void onLoaded(String s) {

    }

    @Override
    public void onAdStarted() {

    }

    @Override
    public void onVideoStarted() {

    }

    @Override
    public void onVideoEnded() {

    }

    @Override
    public void onError(ErrorReason errorReason) {

    }

};

private PlaybackEventListener playbackEventListener = new
PlaybackEventListener() {
    @Override
    public void onPlaying() {

    }

    @Override
    public void onPaused() {

    }
}

```

```

@Override
public void onStoped() {

}

@Override
public void onBuffering(boolean b) {

}

@Override
public void onSeekTo(int i) {

}

};

@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);

    if(newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE)
    {
        if (mPlayer != null)
            mPlayer.setFullscreen(true);
    }else if (newConfig.orientation ==
Configuration.ORIENTATION_PORTRAIT)
    {
        if (mPlayer != null)
            mPlayer.setFullscreen(false);
    }

}

@Override
public void onFullscreen(boolean fullsize) {
    if (fullsize)
    {
        setRequestedOrientation(LANDSCAPE_ORIENTATION);
    }else
    {
        setRequestedOrientation(PORTRAIT_ORIENTATION);
    }
}

@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {

    if (keyCode == KeyEvent.KEYCODE_VOLUME_UP) {
        AudioManager audioManager = (AudioManager)
getBaseContext().getSystemService(
        Context.AUDIO_SERVICE);
        audioManager.adjustStreamVolume(AudioManager.STREAM_MUSIC,
        AudioManager.ADJUST_RAISE,
        AudioManager.FLAG_REMOVE_SOUND_AND_VIBRATE |
AudioManager.FLAG_SHOW_UI);
        return true;
    }
    else if (keyCode == KeyEvent.KEYCODE_VOLUME_DOWN) {
        AudioManager audioManager = (AudioManager)
getBaseContext().getSystemService(
        Context.AUDIO_SERVICE);

```

```

        audioManager.adjustStreamVolume(AudioManager.STREAM_MUSIC,
            AudioManager.ADJUST_LOWER,
            AudioManager.FLAG_REMOVE_SOUND_AND_VIBRATE |
AudioManager.FLAG_SHOW_UI);
        return true;
    }

    return super.onKeyDown(keyCode, event);
}

@Override
public void onLoading() {

}

@Override
public void onLoaded(String s) {

}

@Override
public void onAdStarted() {

}

@Override
public void onVideoStarted() {

}

@Override
public void onVideoEnded() {

}

@Override
public void onError(ErrorReason errorReason) {

}
}

```

- **Quiz Activity**

```

package com.rku.meli;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.res.ColorStateList;
import android.graphics.Color;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.text.Html;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;

```

```

import android.widget.Toast;

import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.muddzdev.styleabletoastlibrary.StyleableToast;
import com.rku.meli.SQLite.QuizDbHelper;
import com.rku.meli.SQLite.Session;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

/**
 * Created by Korio Utoro on 9/27/2018.
 */

public class QuizActivity extends Activity {

    private TextView tv_judul_tab, tv_timer, tv_nomor_soal, tv_soal;
    private RadioGroup rbGroup;
    private RadioButton rb1, rb2, rb3, rb4;
    private Button btn_confirm;

    private List<ModelQuiz> questionList;
    private List<ModelAnswer> answerList;
    private List<ModelAnswer> answers;
    private int questionCounter;
    private int questionCountTotal;
    private ModelQuiz currentQuestion;

    JSONObject jawaban;
    JSONArray arrAnswers;

    private int answerCounter;
    private int answerCountTotal;
    private ModelAnswer modelAnswer;

    private QuizDbHelper dbHelper;

    private Session session;

    private int score;
    private boolean answered;

    private static final long COUNTDOWN_IN_MILLIS = 60000;

    private ColorStateList textColorDefaultCd;
    private CountdownTimer countDownTimer;
    private long timeLeftInMillis;

    private String userID, resultID, judul, quizID;
    private int durasi, minNilai;

    // back button pressed
    private long backPressedTime;
    private Toast backToast;

```

```

ProgressDialog progressDialog;

@Override
public void onBackPressed() {
    if (backPressedTime + 2000 > System.currentTimeMillis())
    {
        backToast.cancel();
        konfirmasiKeluar();

    }else
    {
        backToast = Toast.makeText(getApplicationContext(), "Tekan
tombol back sekali lagi untuk keluar", Toast.LENGTH_LONG);
        backToast.show();
    }

    backPressedTime = System.currentTimeMillis();
}

private void konfirmasiKeluar()
{
    new AlertDialog.Builder(QuizActivity.this)
        .setTitle("Anda Yakin Mau Keluar?")
        .setMessage("Anda akan keluar dari Quiz ini.")
        .setCancelable(false)
        .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface,
int i) {
                startActivity(new Intent(QuizActivity.this,
DashboardActivity.class));
            }
        })
        .setNegativeButton("BATAL", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface,
int i) {
                dialogInterface.cancel();
            }
        })
        .show();
}

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

    tv_judul_tab = (TextView) findViewById(R.id.quiz_judul_tab);
    tv_timer = (TextView) findViewById(R.id.quiz_timer);
    tv_nomor_soal = (TextView) findViewById(R.id.quiz_nomor_soal);
    tv_soal = (TextView) findViewById(R.id.quiz_soal);
    rbGroup = (RadioGroup) findViewById(R.id.quiz_radio_group);
    rb1 = (RadioButton) findViewById(R.id.quiz_option1);
    rb2 = (RadioButton) findViewById(R.id.quiz_option2);
    rb3 = (RadioButton) findViewById(R.id.quiz_option3);
    rb4 = (RadioButton) findViewById(R.id.quiz_option4);
    btn_confirm = (Button) findViewById(R.id.quiz_btn_confirm);

    progressDialog = new ProgressDialog(QuizActivity.this);

```

```

        session = new Session(this);
        dbHelper = new QuizDbHelper(this);
        questionList = dbHelper.getAllQuestions();
        questionCountTotal = questionList.size();
        Log.d("JUMLAH SOAL", String.valueOf(questionCountTotal));

        answers = new ArrayList<>();

        showNextQuestion();

        btn_confirm.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(!answered)
                {
                    if (rb1.isChecked() || rb2.isChecked() ||
rb3.isChecked() || rb4.isChecked())
                    {
                        checkAnswer();
                    }else
                    {
                        StyleableToast.makeText(QuizActivity.this,"Silakan
pilih jawaban yang menurut Anda benar", R.style.warningToast).show();
                    }
                }else
                {
                    showNextQuestion();
                }
            }
        });

        textColorDefaultCd = tv_timer.getTextColors();

        // get session quiz
        userID = session.getIdLogin();
        resultID = session.getResultIDQuiz();
        judul = session.getJudulQuiz().toUpperCase();
        durasi = Integer.valueOf(session.getDurasiQuiz());
        minNilai = Integer.valueOf(session.getMinimumNilaiQuiz());

        quizID = session.getIdQuiz();

        tv_judul_tab.setText(judul);

        timeLeftInMillis = durasi * COUNTDOWN_IN_MILLIS;

        startCountDown();
    }

    private void showNextQuestion()
    {
        rbGroup.clearCheck();

        if (questionCounter < questionCountTotal)
        {
            currentQuestion = questionList.get(questionCounter);

            tv_soal.setText(Html.fromHtml(currentQuestion.getSoal()));
            rb1.setText(currentQuestion.getOpsil1());
            rb2.setText(currentQuestion.getOpsil2());

```

```

        rb3.setText(currentQuestion.getOpsi3());
        rb4.setText(currentQuestion.getOpsi4());

        questionCounter++;
        tv_nomor_soal.setText("Soal : " + questionCounter);
        answered = false;
    }else
    {
        progressDialog.setTitle("Mengirim Jawaban");
        progressDialog.setMessage("Harap tunggu, sedang proses pengiriman...");
        progressDialog.setCancelable(false);
        progressDialog.show();
        checkResult();
    }
}

private void finishQuiz()
{
    finish();
}

private void startCountDown()
{
    countDownTimer = new CountDownTimer(timeLeftInMillis, 1000) {
        @Override
        public void onTick(long millisUntilFinished) {
            timeLeftInMillis = millisUntilFinished;
            updateCountDownText();
        }

        @Override
        public void onFinish() {
            timeLeftInMillis = 0;
            updateCountDownText();
            progressDialog.setTitle("Waktu habis");
            progressDialog.setMessage("Harap tunggu, sedang proses pengiriman...");
            progressDialog.setCancelable(false);
            progressDialog.show();
            checkResult();
        }
    }.start();
}

private void checkResult()
{
    jawaban = new JSONObject();
    arrAnswers = new JSONArray();
    answerList = dbHelper.getAllAnswers();
    answerCountTotal = answerList.size();
    JSONObject [] jsonAnswer = new JSONObject[answerList.size()];
    for (int i = 0; i < answerCountTotal; i++)
    {
        jsonAnswer[i] = new JSONObject();
        modelAnswer = answerList.get(i);
        Log.d("ANSWERS QID", modelAnswer.getQid());
        Log.d("ANSWERS AID", modelAnswer.getOption_id());
        Log.d("ANSWERS SCORE", String.valueOf(modelAnswer.getScore()));
        try {
            jsonAnswer[i].put("qid", modelAnswer.getQid());

```

```

        jsonAnswer[i].put("q_option", modelAnswer.getOption_id());
        jsonAnswer[i].put("score_u", modelAnswer.getScore());
        arrAnswers.put(jsonAnswer[i]);
    } catch (JSONException e) {
        e.printStackTrace();
    }

}

try {
    jawaban.put("jawaban", arrAnswers);
} catch (JSONException e) {
    e.printStackTrace();
}

Log.d("ANSWER JSON", String.valueOf(jawaban));

if (score < minNilai)
{
    Log.d("HASIL QUIZ", "NILAI KURANG");
    progressDialog.cancel();
    new AlertDialog.Builder(QuizActivity.this)
        .setTitle("HASIL QUIZ")
        .setMessage("Nilai Anda masih kurang dari batas nilai
minimum!")
        .setCancelable(false)
        .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface
dialogInterface, int i) {
                submitQuiz(resultID, userID, arrAnswers);
            }
        }).show();
}
else
{
    Log.d("HASIL QUIZ", "NILAI CUKUP");
    progressDialog.cancel();
    new AlertDialog.Builder(QuizActivity.this)
        .setTitle("HASIL QUIZ")
        .setMessage("Selamat, Anda berhasil melampaui batas
nilai minimum!")
        .setCancelable(false)
        .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface
dialogInterface, int i) {
                submitQuiz(resultID, userID, arrAnswers);
            }
        }).show();
    }
}

private void submitQuiz(String resultID, final String userID, JSONArray
jawaban)
{
    String url_submit_quiz =
"https://meli.voidsolution.com/api/submit_quiz_api/";

    JSONObject params = new JSONObject();

```

```

        try {
            params.put("result_id", Integer.valueOf(resultID));
            params.put("user_id", Integer.valueOf(userID));
            params.put("jawaban", jawaban);
        } catch (JSONException e) {
            e.printStackTrace();
        }

        Log.d("JAWABAN DISUBMIT", jawaban.toString());

        JsonObjectRequest requestSubmit = new
        JsonObjectRequest(url_submit_quiz, params, new
        Response.Listener<JsonObject>() {
            @Override
            public void onResponse(JsonObject response) {
                progressDialog.dismiss();
                try {
                    int stats = response.getInt("status");
                    if (stats == 0)
                    {
                        finishQuiz();
                        StyleableToast.makeText(QuizActivity.this, "Terjadi
kesalahan, jawaban Anda gagal disimpan!", R.style.warningToast).show();
                    }else
                    {
                        StyleableToast.makeText(QuizActivity.this, "Quiz
telah selesai dikerjakan, Jawaban Anda telah tersimpan!",
R.style.updatedToast).show();
                        session.setBeresQuiz(true);
                        refreshResult(userID, quizID);
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {

            }
        }));

        MySingleton.getInstance(getApplicationContext()).addToRequestQueue(requestSubmit);
    }

    private void refreshResult(String userID, String quizID)
    {
        String url_get_quiz =
        "https://meli.voidsolution.com/api/get_quiz_api/";

        JsonObject params = new JsonObject();
        try {
            params.put("user_id", Integer.valueOf(userID));
            params.put("quiz_id", Integer.valueOf(quizID));
        } catch (JSONException e) {
            e.printStackTrace();
        }

        JsonObjectRequest requestQuiz = new JsonObjectRequest(url_get_quiz,
        params, new Response.Listener<JsonObject>() {

```

```

        @Override
        public void onResponse(JSONObject response) {
            try {
                int status_getQuiz = response.getInt("status");

                if (status_getQuiz == 0)
                {
                    String msg_getQuiz = response.getString("msg");

                    new AlertDialog.Builder(QuizActivity.this)
                        .setTitle("Oops")
                        .setMessage(msg_getQuiz)
                        .setCancelable(false)
                        .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface
dialogInterface, int i) {
                                dialogInterface.cancel();
                            }
                        }).show();
                }
                }else
                {
                    JSONArray arrResult =
response.getJSONArray("user_result");
                    Log.d("RESULT TERBARU", arrResult.toString());
                    session.setArrResult(arrResult.toString());
                    progressDialog.dismiss();
                    progressDialog.setTitle("Menyimpan Data");
                    progressDialog.setMessage("Harap tunggu, sedang
proses penyimpanan...");
                    progressDialog.setCancelable(false);
                    progressDialog.show();
                    finishQuiz();
                }

                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {

            }

        });

        MySingleton.getInstance(QuizActivity.this).addToRequestQueue(requestQuiz);
    }

    private void checkAnswer()
    {
        answered = true;
        modelAnswer = new ModelAnswer();

        RadioButton rbSelected =
findViewById(rbGroup.getCheckedRadioButtonId());
        int answerNr = rbGroup.indexOfChild(rbSelected) + 1;

        modelAnswer.setQid(currentQuestion.getQuestion_id());
    }

```

```

        if (answerNr == 1)
        {
            modelAnswer.setOption_id(currentQuestion.getId_opsi1());
        }
        else if (answerNr == 2)
        {
            modelAnswer.setOption_id(currentQuestion.getId_opsi2());
        }
        else if (answerNr == 3)
        {
            modelAnswer.setOption_id(currentQuestion.getId_opsi3());
        }
        else if (answerNr == 4)
        {
            modelAnswer.setOption_id(currentQuestion.getId_opsi4());
        }

        if ((answerNr - 1) == currentQuestion.getNomorJawaban())
        {
            score += 20;
            modelAnswer.setScore(1);
        }else
        {
            score += 0;
            modelAnswer.setScore(0);
        }

        dbHelper.addAnswer(modelAnswer);
        answers.add(modelAnswer);
        showNextQuestion();
    }

    private void updateCountDownText()
    {
        int minutes = (int) (timeLeftInMillis / 1000) / 60;
        int seconds = (int) (timeLeftInMillis / 1000) % 60;

        String timeFormatted = String.format(Locale.getDefault(),
"%02d:%02d", minutes, seconds);

        tv_timer.setText("Waktu: " + timeFormatted);

        if (timeLeftInMillis < 10000)
        {
            tv_timer.setTextColor(Color.RED);
        }else
        {
            tv_timer.setTextColor(textColorDefaultCd);
        }
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        if (countDownTimer != null)
        {
            countDownTimer.cancel();
        }
    }
}

```