LAMPIRAN

KODING

PROGRAM

```
package com.enwif.plwisata.cnt;


import android.app.DatePickerDialog;
import android.app.ProgressDialog;
import android.app.TimePickerDialog;
import android.content.Context;
import android.support.constraint.ConstraintLayout;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.ScrollView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.TimePicker;
```

```
import com.android.volley.AuthFailureError;
import com.android.volley.NetworkResponse;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.ServerError;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.HttpHeaderParser;
import com.android.volley.toolbox.StringRequest;
import com.enwif.plwisata.R;
import com.enwif.plwisata.cnn.Cnn;
import com.enwif.plwisata.mdl.MenuMdl;
import com.enwif.plwisata.rqs.RequestRqs;

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

import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
```

```
import java.util.List;
import java.util.Map;

public class Booking extends AppCompatActivity {
    private ScrollView root;
    private TextView tvMeja;
    private EditText etNama, etNoHP, etJmlOrang, etTgl, etJumlah, etPesanan, etTotalBayar, etJam;
    private Button btnCheck, btnSend, btnAddPesanan;
    private Spinner spPesanan;

    private String id, tgl;
    private List<MenuMdl> mdls;

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

        id = getIntent().getStringExtra("id");

        root = (ScrollView) findViewById(R.id.root);
        tvMeja = (TextView) findViewById(R.id.tvMeja);
```

```
etNama = (EditText) findViewById(R.id.etNama);
etNoHP = (EditText) findViewById(R.id.etNoHP);
etJmlOrang = (EditText) findViewById(R.id.etJmlOrang);
etTgl = (EditText) findViewById(R.id.etTgl);
etJumlah = (EditText) findViewById(R.id.etJumlah);
etPesanan = (EditText) findViewById(R.id.etPesanan);
etTotalBayar = (EditText) findViewById(R.id.etTotalBayar);
etJam = (EditText) findViewById(R.id.etJam);
btnCheck = (Button) findViewById(R.id.btnCheck);
btnSend = (Button) findViewById(R.id.btnSend);
btnAddPesanan = (Button) findViewById(R.id.btnAddPesanan);
spPesanan = (Spinner) findViewById(R.id.spPesanan);

btnCheck.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        checkAvailability();
    }
});

btnSend.setOnClickListener(new View.OnClickListener() {
    @Override
```

```
        public void onClick(View v) {
            send();
        }
    });

    etTgl.setKeyListener(null);
    etTgl.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Calendar calendar = Calendar.getInstance();
            int mYear = calendar.get(Calendar.YEAR);
            int mMonth = calendar.get(Calendar.MONTH);
            int mDay = calendar.get(Calendar.DAY_OF_MONTH);

            DatePickerDialog datePickerDialog = new
DatePickerDialog(Booking.this, new DatePickerDialog.OnDateSetListener() {
                @Override
                public void onDateSet(DatePicker view, int year, int month, int
dayOfMonth) {
                    etTgl.setText(String.format("%02d",dayOfMonth) + "-" +
String.format("%02d",(month + 1)) + "-" + year);
                    tgl = year + "-" + String.format("%02d",(month + 1)) + "-" +
String.format("%02d",dayOfMonth);
                }
```

```
            }, mYear, mMonth, mDay);


            datePickerDialog.show();
        }
    });

    etJam.setKeyListener(null);
    etJam.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Calendar calendar = Calendar.getInstance();
            int hour = calendar.get(Calendar.HOUR_OF_DAY);
            int minute = calendar.get(Calendar.MINUTE);


            TimePickerDialog timePickerDialog = new
TimePickerDialog(Booking.this, new TimePickerDialog.OnTimeSetListener() {
                @Override
                public void onTimeSet(TimePicker view, int hourOfDay, int minute)
{
                    etJam.setText(String.format("%02d", hourOfDay) + ":" +
String.format("%02d", minute));
                }
            }, hour, minute, true);
```

```
            timePickerDialog.show();
        }
    });

    etTotalBayar.setKeyListener(null);

    btnAddPesanan.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String s = etPesanan.getText().toString();
            int t;
            if(etTotalBayar.getText().toString().equals("")) {
                t = 0;
            } else {
                t = Integer.parseInt(etTotalBayar.getText().toString());
            }
            MenuMdl m = mdls.get(spPesanan.getSelectedItemPosition());
            int h = Integer.parseInt(etJumlah.getText().toString()) *
Integer.parseInt(m.getHarga());
            if(s.equals("")) {
                etPesanan.setText(m.getNama() + "(" + etJumlah.getText().toString()
+ " x Rp." + m.getHarga() + " = Rp. " + h + ")");
```

```
                etTotalBayar.setText(Integer.toString(h));
            } else {
                etPesanan.setText(s + ",\n" + spPesanan.getSelectedItem().toString()
+ "(" + etJumlah.getText().toString() + " x Rp." + m.getHarga() + " = Rp. " + h +
")");
                etTotalBayar.setText(Integer.toString(t + h));
            }
        }
    });

    etPesanan.setKeyListener(null);
    etPesanan.setOnLongClickListener(new View.OnLongClickListener() {
        @Override
        public boolean onLongClick(View v) {
            etPesanan.setText("");
            etTotalBayar.setText("");
            return false;
        }
    });

    checkAvailability();

    setDt();
```

```
    }

    private void setDt() {
        mdls = new ArrayList<>();
        final List<String> list = new ArrayList<>();

        StringRequest stringRequest = new StringRequest(Request.Method.POST,
Cnn.url + "menu/some", new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    JSONArray jsonArray = new JSONArray(response);
                    for (int i = 0; i < jsonArray.length(); i++) {
                        JSONObject jsonObject = jsonArray.getJSONObject(i);

                        MenuMdl mdl = new MenuMdl();
                        mdl.setIdmenu(jsonObject.getString("idmenu"));
                        mdl.setNama(jsonObject.getString("nama"));
                        mdl.setHarga(jsonObject.getString("harga"));

                        mdls.add(mdl);
                        list.add(jsonObject.getString("nama"));
                    }
```

```
                ArrayAdapter<String> adp = new ArrayAdapter<>(Booking.this,
android.R.layout.simple_list_item_1, list);

adp.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_ite
m);
                spPesanan.setAdapter(adp);
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Log.e("setDt", error.toString()); // Or for detail use below
            NetworkResponse response = error.networkResponse;
            if (error instanceof ServerError && response != null) {
                try {
                    String res = new String(response.data,
HttpHeaderParser.parseCharset(response.headers, "utf-8"));
                    // Now you can use any deserializer to make sense of data
                    Log.e("setDt", res);
                } catch (UnsupportedEncodingException e1) {
                    // Couldn't properly decode data to string
```

```
                    e1.printStackTrace();
                }
            }
        }
    }) {
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> params = new HashMap<>();
            params.put("wskuliner_idwskuliner", id);
            return params;
        }
    };

    RequestRqs.getInstance().addToRequestQueue(stringRequest);
}

private void checkAvailability() {
    final ProgressDialog progressDialog = ProgressDialog.show(Booking.this,
getString(getResources().getIdentifier("strInfLoadData", "string",
getPackageName())), getString(getResources().getIdentifier("strInfWait", "string",
getPackageName())), false, false);

    StringRequest stringRequest = new StringRequest(Request.Method.POST,
Cnn.url + "booking/get", new Response.Listener<String>() {
```

```
@Override
public void onResponse(String response) {
    progressDialog.dismiss();

    try {
        JSONObject jsonObject = new JSONObject(response);
        if(jsonObject.getString("meja").equalsIgnoreCase("Tersedia")) {
            tvMeja.setText("Tersedia");
            etNama.setEnabled(true);
            etNoHP.setEnabled(true);
            etJmlOrang.setEnabled(true);
            btnSend.setEnabled(true);
        } else {
            tvMeja.setText("Tidak");
            etNama.setEnabled(false);
            etNoHP.setEnabled(false);
            etJmlOrang.setEnabled(false);
            btnSend.setEnabled(false);
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
```

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


                Log.e("setDt", error.toString()); // Or for detail use below
                NetworkResponse response = error.networkResponse;
                if (error instanceof ServerError && response != null) {
                    try {
                        String res = new String(response.data,
HttpHeaderParser.parseCharset(response.headers, "utf-8"));
                        // Now you can use any deserializer to make sense of data
                        Log.e("setDt", res);
                    } catch (UnsupportedEncodingException e1) {
                        // Couldn't properly decode data to string
                        e1.printStackTrace();
                    }
                }
            }
        }) {
            @Override
            protected Map<String, String> getParams() throws AuthFailureError {
```

```
            Map<String, String> params = new HashMap<>();
            params.put("idwskuliner", id);
            return params;
        }
    };

    RequestRqs.getInstance().addToRequestQueue(stringRequest);
}

private void send() {
    final ProgressDialog progressDialog = ProgressDialog.show(Booking.this, getString(getResources().getIdentifier("strInfSending", "string", getPackageName())), getString(getResources().getIdentifier("strInfWait", "string", getPackageName())), false, false);

    StringRequest stringRequest = new StringRequest(Request.Method.POST, Cnn.url + "booking/ins", new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            progressDialog.dismiss();

            if(response.equalsIgnoreCase("1")) {
```

```
                Snackbar.make(root,
getString(getResources().getIdentifier("strInfSuccessSend", "string",
getPackageName())), Snackbar.LENGTH_LONG).show();
                showInfo();
            } else {
                Snackbar.make(root,
getString(getResources().getIdentifier("strInfFailSend", "string",
getPackageName())), Snackbar.LENGTH_LONG).show();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            progressDialog.dismiss();

            Log.e("setDt", error.toString()); // Or for detail use below
            NetworkResponse response = error.networkResponse;
            if (error instanceof ServerError && response != null) {
                try {
                    String res = new String(response.data,
HttpHeaderParser.parseCharset(response.headers, "utf-8"));
                    // Now you can use any deserializer to make sense of data
                    Log.e("setDt", res);
                } catch (UnsupportedEncodingException e1) {
```

```
                // Couldn't properly decode data to string
                e1.printStackTrace();
            }
        }
    }
}) {
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<>();
        params.put("nama", etNama.getText().toString());
        params.put("nohp", etNoHP.getText().toString());
        params.put("jmlorang", etJmlOrang.getText().toString());
        params.put("tgl", tgl);
        params.put("jam", etJam.getText().toString());
        params.put("pesanan", etPesanan.getText().toString());
        params.put("wskuliner_idwskuliner", id);
        return params;
    }
};

RequestRqs.getInstance().addToRequestQueue(stringRequest);
}
```

```
    private void showInfo() {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        LayoutInflater layoutInflater = (LayoutInflater)
this.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View view = layoutInflater.inflate(R.layout.booking_info, null);
        builder.setView(view);
        builder.show();
    }
}
```