

LAMPIRAN

AyamActivity.java

```
package
com.aplikasi.apliaksiternaktelur.admin.aya
m;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;
import
androidx.recyclerview.widget.LinearLayout
tManager;
import
androidx.recyclerview.widget.RecyclerVie
w;

import android.content.Intent;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import
com.aplikasi.apliaksiternaktelur.MainActiv
ity;
import com.aplikasi.apliaksiternaktelur.R;
import
com.aplikasi.apliaksiternaktelur.admin.telur.table_telur;
import
com.google.firebase.database.DataSnapsho
t;
import
com.google.firebase.database.DatabaseErr
or;
import
com.google.firebase.database.DatabaseRef
erence;
import
com.google.firebase.database.FirebaseData
base;
```

```
import
com.google.firebase.database.ValueEventL
istener;

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

public class AyamActivity extends
AppCompatActivity {

    private RecyclerView recyclerView;
    private AyamAdapter ayamAdapter;
    private List<DataAyam> ayamList;
    private List<DataAyam> filteredList; //
Untuk menyimpan hasil filter

    private DatabaseReference
databaseReference;
    private TextView textViewCount;
    private EditText editTextSearch;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.data_ayam);

        textViewCount =
findViewById(R.id.textViewCount); //
Inisialisasi textViewCount
        editTextSearch =
findViewById(R.id.cariKondisi); //
Inisialisasi editTextSearch

        recyclerView =
findViewById(R.id.recyclerViewAy);
        recyclerView.setLayoutManager(new
LinearLayoutManager(this));

        ayamList = new ArrayList<>();
        filteredList = new ArrayList<>(); //
Inisialisasi filteredList
        ayamAdapter = new
AyamAdapter(this, ayamList); //
Menambahkan context ke konstruktor
```

AyamAdapter

```
recyclerView.setAdapter(ayamAdapter);

// Inisialisasi referensi database
Firebase
    databaseReference =
FirebaseDatabase.getInstance().getReferen
ce().child("admin/ayam");

// Mengambil data dari Firebase
fetchDataFromFirebase();

// Set listener untuk EditText
pencarian

editTextSearch.addTextChangedListener(n
ew TextWatcher() {
    @Override
    public void
beforeTextChanged(CharSequence s, int
start, int count, int after) {
        // Tidak melakukan apa-apa
    }

    @Override
    public void
onTextChanged(CharSequence s, int start,
int before, int count) {
        // Filter data berdasarkan teks
input
        filterData(s.toString());
    }

    @Override
    public void
afterTextChanged(Editable s) {
        // Tidak melakukan apa-apa
    }
});

// Metode untuk mengambil data dari
```

Firestore

```
private void fetchDataFromFirestore() {

databaseReference.addValueEventListener
(new ValueEventListener() {
    @Override
    public void
onDataChange(@NonNull DataSnapshot
snapshot) {
        ayamList.clear();
        int totalAyam = 0; // Variabel
untuk menyimpan total jumlahAyam

        for (DataSnapshot dataSnapshot
: snapshot.getChildren()) {
            // Mengambil data dari
Firestore
            String id =
dataSnapshot.child("id").getValue(String.c
lass);

            String jenis =
dataSnapshot.child("jenis").getValue(Strin
g.class);

            int jumlahAyam =
dataSnapshot.child("jumlahAyam").getVal
ue(Integer.class);

            String kandangAyam =
dataSnapshot.child("kandangAyam").getV
alue(String.class);

            String status =
dataSnapshot.child("status").getValue(Stri
ng.class);

            String tanggal =
dataSnapshot.child("tanggal").getValue(Str
ing.class);

            // Membuat objek DataAyam
            DataAyam dataAyam = new
DataAyam(id, jenis, jumlahAyam,
kandangAyam, status, tanggal);

            // Hanya tambahkan
jumlahAyam ke totalAyam jika statusnya
"hidup" atau "sakit"
            if
```

```

(status.equalsIgnoreCase("Sehat") ||
status.equalsIgnoreCase("Sakit")) {
    totalAyam += jumlahAyam;
}

// Tambahkan dataAyam ke
ayamList
    ayamList.add(dataAyam);
}

// Memperbarui textViewCount
dengan totalAyam
    textViewCount.setText("Total
Ayam: " + totalAyam);

// Memberitahu adapter bahwa
data telah berubah
ayamAdapter.notifyDataSetChanged();
}

@Override
public void
onCancelled(@NonNull DatabaseError
error) {
    // Menangani kesalahan database

    Toast.makeText(AyamActivity.this, "Gagal
mengambil data: " + error.getMessage(),
    Toast.LENGTH_SHORT).show();
}
});
}

```

```

// Metode untuk memfilter data
berdasarkan query
private void filterData(String query) {
    filteredList.clear();
    for (DataAyam data : ayamList) {
        if
(data.getStatus().toLowerCase().contains(q
uery.toLowerCase())) {
            filteredList.add(data);
        }
    }
    ayamAdapter.filterList(filteredList);
}

```

```

}

// Metode untuk menambah ayam
public void tambah_ayam(View view) {
    Intent intent = new Intent(this,
TambahAyam.class);
    startActivity(intent);
    finish();
}

// Metode untuk kembali ke aktivitas
utama
public void pressback(View view) {
    Intent intent = new Intent(this,
MainActivity.class);
    startActivity(intent);
    finish();
} public void btn_detailAyam(View
view) {
    Intent intent = new Intent(this,
table_ayam.class);
    startActivity(intent);
    finish();
}

}

```

AyamAdapter.java

```

package
com.aplikasi.apliksiternaktelur.admin.aya
m;

```

```

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

```

```

import androidx.annotation.NonNull;
import
androidx.recyclerview.widget.RecyclerVie

```

```

w;

import com.aplikasi.apliaksiternaktelur.R;

import java.util.List;

public class AyamAdapter extends
RecyclerView.Adapter<RecyclerView.Vie
wHolder> {
    private static final int
VIEW_TYPE_SEHAT = 1;
    private static final int
VIEW_TYPE_SAKIT = 2;
    private static final int
VIEW_TYPE_MATI = 3;

    private Context context;
    private List<DataAyam> ayamList;

    public AyamAdapter(Context context,
List<DataAyam> ayamList) {
        this.context = context;
        this.ayamList = ayamList;
    }

    @Override
    public int getItemViewType(int
position) {
        switch
(ayamList.get(position).getStatus()) {
            case "Sehat":
                return VIEW_TYPE_SEHAT;
            case "Sakit":
                return VIEW_TYPE_SAKIT;
            case "Mati":
                return VIEW_TYPE_MATI;
            default:
                return -1;
        }
    }

    @NonNull
    @Override
    public RecyclerView.ViewHolder
onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {

```

```

        LayoutInflater inflater =
LayoutInflater.from(parent.getContext());
        View itemView;
        RecyclerView.ViewHolder
viewHolder = null;

        switch (viewType) {
            case VIEW_TYPE_SEHAT:
                itemView =
inflater.inflate(R.layout.item_ayam, parent,
false);
                viewHolder = new
AyamSehatViewHolder(itemView);
                break;
            case VIEW_TYPE_SAKIT:
                itemView =
inflater.inflate(R.layout.item_ayam_sakit,
parent, false);
                viewHolder = new
AyamSakitViewHolder(itemView);
                break;
            case VIEW_TYPE_MATI:
                itemView =
inflater.inflate(R.layout.item_ayam_mati,
parent, false);
                viewHolder = new
AyamMatiViewHolder(itemView);
                break;
            default:
                throw new
IllegalArgumentException("Invalid view
type");
        }

        return viewHolder;
    }

    @Override
    public void
onBindViewHolder(@NonNull
RecyclerView.ViewHolder holder, int
position) {
        DataAyam dataAyam =
ayamList.get(position);

        switch (holder.getItemViewType()) {

```

```

        case VIEW_TYPE_SEHAT:
            ((AyamSehatViewHolder)
holder).bind(dataAyam);
            break;
        case VIEW_TYPE_SAKIT:
            ((AyamSakitViewHolder)
holder).bind(dataAyam);
            break;
        case VIEW_TYPE_MATI:
            ((AyamMatiViewHolder)
holder).bind(dataAyam);
            break;
        default:
            throw new
IllegalArgumentException("Invalid view
type");
    }
}

@Override
public int getItemCount() {
    return ayamList.size();
}

public class AyamSehatViewHolder
extends RecyclerView.ViewHolder {
    TextView textViewKandang,
textViewDateL, textViewChickenType,
textViewAmount, textViewKondisi;
    ImageButton btnEdit;

    AyamSehatViewHolder(View
itemView) {
        super(itemView);
        textViewKandang =
itemView.findViewById(R.id.textViewKan
dang);
        textViewDateL =
itemView.findViewById(R.id.textViewDat
eL);
        textViewChickenType =
itemView.findViewById(R.id.textViewChic
kenType);
        textViewAmount =
itemView.findViewById(R.id.textViewAmo
unt);

```

```

        textViewKondisi =
itemView.findViewById(R.id.textViewKon
disi);
        btnEdit =
itemView.findViewById(R.id.btnEdit);

        btnEdit.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                int position =
getAdapterPosition();
                if (position !=
RecyclerView.NO_POSITION) {
                    DataAyam dataAyam =
ayamList.get(position);
                    Intent intent = new
Intent(context, Edit_Ayam.class);
                    intent.putExtra("idAyam",
dataAyam.getId());
                    intent.putExtra("jenis",
dataAyam.getJenis());

                    intent.putExtra("jumlahAyam",
dataAyam.getJumlahAyam());

                    intent.putExtra("kandangAyam",
dataAyam.getKandangAyam());
                    intent.putExtra("status",
dataAyam.getStatus());
                    intent.putExtra("tanggal",
dataAyam.getTanggal());

                    context.startActivity(intent);
                } else {
                    Toast.makeText(context,
"Invalid position",
Toast.LENGTH_SHORT).show();
                }
            }
        });

        void bind(DataAyam dataAyam) {
            textViewKandang.setText(dataAyam.getKa

```

```

    kandangAyam());

    textViewDateL.setText(dataAyam.getTang
gal());

    textViewChickenType.setText(dataAyam.g
etJenis());

    textViewAmount.setText(String.valueOf(d
ataAyam.getJumlahAyam()));

    textViewKondisi.setText(dataAyam.getStat
us());
    }
}

```

```

    public class AyamSakitViewHolder
extends RecyclerView.ViewHolder {
        TextView textViewKandang,
        textViewDateL, textViewChickenType,
        textViewAmount, textViewKondisi;
        ImageButton btnEdit;

        AyamSakitViewHolder(View
itemView) {
            super(itemView);
            textViewKandang =
itemView.findViewById(R.id.textViewKan
dang);
            textViewDateL =
itemView.findViewById(R.id.textViewDat
eL);
            textViewChickenType =
itemView.findViewById(R.id.textViewChic
kenType);
            textViewAmount =
itemView.findViewById(R.id.textViewAmo
unt);
            textViewKondisi =
itemView.findViewById(R.id.textViewKon
disi);
            btnEdit =
itemView.findViewById(R.id.btnEdit);

            btnEdit.setOnClickListener(new
View.OnClickListener() {

```

```

                @Override
                public void onClick(View v) {
                    int position =
getAdapterPosition();
                    if (position !=
RecyclerView.NO_POSITION) {
                        DataAyam dataAyam =
ayamList.get(position);
                        Intent intent = new
Intent(context, Edit_Ayam.class);
                        intent.putExtra("idAyam",
dataAyam.getId());
                        intent.putExtra("jenis",
dataAyam.getJenis());

                        intent.putExtra("jumlahAyam",
dataAyam.getJumlahAyam());

                        intent.putExtra("kandangAyam",
dataAyam.getKandangAyam());
                        intent.putExtra("status",
dataAyam.getStatus());
                        intent.putExtra("tanggal",
dataAyam.getTanggal());

                        context.startActivity(intent);
                    } else {
                        Toast.makeText(context,
"Invalid position",
Toast.LENGTH_SHORT).show();
                    }
                }
            });
        }

        void bind(DataAyam dataAyam) {

            textViewKandang.setText(dataAyam.getKa
ndangAyam());

            textViewDateL.setText(dataAyam.getTang
gal());

            textViewChickenType.setText(dataAyam.g
etJenis());

```

```
textViewAmount.setText(String.valueOf(d
ataAyam.getJumlahAyam()));
```

```
textViewKondisi.setText(dataAyam.getStat
us());
    }
}
```

```
public class AyamMatiViewHolder
extends RecyclerView.ViewHolder {
    TextView textViewKandang,
    textViewDateL, textViewChickenType,
    textViewAmount, textViewKondisi;
    ImageButton btnEdit;
```

```
    AyamMatiViewHolder(View
itemView) {
        super(itemView);
        textViewKandang =
itemView.findViewById(R.id.textViewKan
dang);
        textViewDateL =
itemView.findViewById(R.id.textViewDat
eL);
        textViewChickenType =
itemView.findViewById(R.id.textViewChic
kenType);
        textViewAmount =
itemView.findViewById(R.id.textViewAmo
unt);
        textViewKondisi =
itemView.findViewById(R.id.textViewKon
disi);
        btnEdit =
itemView.findViewById(R.id.btnEdit);
```

```
        btnEdit.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                int position =
getAdapterPosition();
                if (position !=
RecyclerView.NO_POSITION) {
                    DataAyam dataAyam =
ayamList.get(position);
```

```
                Intent intent = new
Intent(context, Edit_Ayam.class);
                intent.putExtra("idAyam",
dataAyam.getId());
                intent.putExtra("jenis",
dataAyam.getJenis());
```

```
                intent.putExtra("jumlahAyam",
dataAyam.getJumlahAyam());

                intent.putExtra("kandangAyam",
dataAyam.getKandangAyam());
                intent.putExtra("status",
dataAyam.getStatus());
                intent.putExtra("tanggal",
dataAyam.getTanggal());
```

```
                context.startActivity(intent);
            } else {
                Toast.makeText(context,
"Invalid position",
Toast.LENGTH_SHORT).show();
            }
        }
    });
}
```

```
void bind(DataAyam dataAyam) {
```

```
    textViewKandang.setText(dataAyam.getKa
ndangAyam());
```

```
    textViewDateL.setText(dataAyam.getTang
gal());
```

```
    textViewChickenType.setText(dataAyam.g
etJenis());
```

```
    textViewAmount.setText(String.valueOf(d
ataAyam.getJumlahAyam()));
```

```
    textViewKondisi.setText(dataAyam.getStat
us());
    }
}
```

```

        public void filterList(List<DataAyam>
filteredList) {
            ayamList = filteredList;
            notifyDataSetChanged();
        }
    }
}

```

DataAyam.java

```

package
com.aplikasi.apliaksiternaktelur.admin.aya
m;

```

```

public class DataAyam {
    private String id;
    private String jenis;
    private int jumlahAyam;
    private String kandangAyam;
    private String status;
    private String tanggal;

    // Constructor
    public DataAyam(String id, String jenis,
int jumlahAyam, String kandangAyam,
String status, String tanggal) {
        this.id = id;
        this.jenis = jenis;
        this.jumlahAyam = jumlahAyam;
        this.kandangAyam = kandangAyam;
        this.status = status;
        this.tanggal = tanggal;
    }

    // Getter dan Setter
    public String getId() { return id; }
    public void setId(String id) { this.id =
id; }

    public String getJenis() { return jenis; }
    public void setJenis(String jenis) {
this.jenis = jenis; }

    public int getJumlahAyam() { return
jumlahAyam; }
    public void setJumlahAyam(int
jumlahAyam) { this.jumlahAyam =

```

```

jumlahAyam; }

```

```

    public String getKandangAyam() {
return kandangAyam; }
    public void setKandangAyam(String
kandangAyam) { this.kandangAyam =
kandangAyam; }

```

```

    public String getStatus() { return status;
}
    public void setStatus(String status) {
this.status = status; }

```

```

    public String getTanggal() { return
tanggal; }
    public void setTanggal(String tanggal) {
this.tanggal = tanggal; }
}

```

Edit_data.java

```

package
com.aplikasi.apliaksiternaktelur.admin.aya
m;

```

```

import
androidx.appcompat.app.AppCompatActivity;

```

```

import android.app.DatePickerDialog;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

```

```

import com.aplikasi.apliaksiternaktelur.R;
import
com.google.firebase.database.DatabaseRef

```

```

erence;
import
com.google.firebase.database.FirebaseData
base;

```

```

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.Locale;

```

```

public class Edit_Ayam extends
AppCompatActivity {

```

```

    private TextView
tanggalMasukTextView;
    private DatePicker datePicker;
    private Spinner editKandang,
editJenisAy, editKondisi;
    private EditText editJumlahAyam;
    private Button btnUpdate, btnHapus,
btnBatal;

```

```

    private DatabaseReference
databaseReference;
    private String idAyam;

```

```

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

```

```

        tanggalMasukTextView =
findViewById(R.id.tanggalMasukTextView
);

```

```

        datePicker =
findViewById(R.id.datePicker);
        editKandang =
findViewById(R.id.editKandang);
        editJenisAy =
findViewById(R.id.editJenisAy);
        editKondisi =
findViewById(R.id.editKondisi);
        editJumlahAyam =
findViewById(R.id.editJumlahAyam);

```

```

        btnUpdate =
findViewById(R.id.btnupdate);
        btnHapus =
findViewById(R.id.btnhapus);
        btnBatal =
findViewById(R.id.btncancel);

```

```

        databaseReference =
FirebaseDatabase.getInstance().getReferen
ce().child("admin/ayam");

```

```

        Bundle extras =
getIntent().getExtras();
        if (extras != null) {
            idAyam =
extras.getString("idAyam");
            String jenis =
extras.getString("jenis");
            int jumlahAyam =
extras.getInt("jumlahAyam");
            String kandangAyam =
extras.getString("kandangAyam");
            String status =
extras.getString("status");
            String tanggal =
extras.getString("tanggal");

```

```

        tanggalMasukTextView.setText(tanggal);
        setDatePicker(tanggal);
        setSpinnerSelections(jenis,
kandangAyam, status);

```

```

        editJumlahAyam.setText(String.valueOf(ju
mlahAyam));
    }

```

```

        setupDatePicker();
        setupSpinners();

```

```

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

```

```

    });

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

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

    private void setDatePicker(String
tanggal) {
        SimpleDateFormat sdf = new
SimpleDateFormat("dd-MM-yyyy",
Locale.US);
        Calendar cal =
Calendar.getInstance();
        try {
            cal.setTime(sdf.parse(tanggal));
            int year = cal.get(Calendar.YEAR);
            int month =
cal.get(Calendar.MONTH);
            int day =
cal.get(Calendar.DAY_OF_MONTH);
            datePicker.updateDate(year, month,
day);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private void setSpinnerSelections(String
jenis, String kandangAyam, String status)
{
        ArrayAdapter<CharSequence>
adapterJenisAy =

```

```

ArrayAdapter.createFromResource(this,
R.array.listJenisAyam,
android.R.layout.simple_spinner_item);

adapterJenisAy.setDropDownViewResourc
e(android.R.layout.simple_spinner_dropdo
wn_item);

editJenisAy.setAdapter(adapterJenisAy);
    if (jenis != null) {
        int position =
adapterJenisAy.getPosition(jenis);
        editJenisAy.setSelection(position);
    }

        ArrayAdapter<CharSequence>
adapterKandang =
ArrayAdapter.createFromResource(this,
R.array.listKandang,
android.R.layout.simple_spinner_item);

adapterKandang.setDropDownViewResou
rce(android.R.layout.simple_spinner_drop
down_item);

editKandang.setAdapter(adapterKandang);
    if (kandangAyam != null) {
        int position =
adapterKandang.getPosition(kandangAya
m);

editKandang.setSelection(position);
    }

        ArrayAdapter<CharSequence>
adapterKondisi =
ArrayAdapter.createFromResource(this,
R.array.listKondisi,
android.R.layout.simple_spinner_item);

adapterKondisi.setDropDownViewResourc
e(android.R.layout.simple_spinner_dropd
own_item);

editKondisi.setAdapter(adapterKondisi);
    if (status != null) {

```

```

        int position =
adapterKondisi.getPosition(status);
        editKondisi.setSelection(position);
    }
}

```

```

private void setupDatePicker() {

tanggalMasukTextView.setOnClickListener(
new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Calendar cal =
Calendar.getInstance();
        int year =
cal.get(Calendar.YEAR);
        int month =
cal.get(Calendar.MONTH);
        int day =
cal.get(Calendar.DAY_OF_MONTH);

```

```

        DatePickerDialog dialog = new
DatePickerDialog(Edit_Ayam.this,
        new
DatePickerDialog.OnDateSetListener() {
            @Override
            public void
onDateSet(DatePicker view, int year, int
monthOfYear, int dayOfMonth) {
                // Update TextView
with selected date
                Calendar selectedDate
= Calendar.getInstance();

selectedDate.set(Calendar.YEAR, year);

selectedDate.set(Calendar.MONTH,
monthOfYear);

selectedDate.set(Calendar.DAY_OF_MON
TH, dayOfMonth);

```

```

                // Format the selected
date to "yyyy-MM-dd"
                SimpleDateFormat sdf
= new SimpleDateFormat("dd-MM-yyyy",

```

```

Locale.US);
                String formattedDate =
sdf.format(selectedDate.getTime());

```

```

                // Set the formatted
date to TextView

```

```

tanggalMasukTextView.setText(formatted
Date);

```

```

                // Update the
DatePicker to reflect the selected date

```

```

datePicker.updateDate(year, monthOfYear,
dayOfMonth);
            }
            }, year, month, day);
            dialog.show();
        }
    });
}

```

```

private void setupSpinners() {

```

```

editJenisAy.setOnItemSelectedListener(ne
w AdapterView.OnItemSelectedListener()
{

```

```

    @Override
    public void
onItemSelected(AdapterView<?> parent,
View view, int position, long id) {
        // Handle when item selected
    }

```

```

    @Override
    public void
onNothingSelected(AdapterView<?>
parent) {
        // Do nothing
    }
});

```

```

editKandang.setOnItemSelectedListener(n

```

```

new AdapterView.OnItemClickListener()
{
    @Override
    public void
onItemSelected(AdapterView<?> parent,
View view, int position, long id) {
        // Handle when item selected
    }

    @Override
    public void
onNothingSelected(AdapterView<?>
parent) {
        // Do nothing
    }
});

```

```

editKondisi.setOnItemClickListener(ne
w AdapterView.OnItemClickListener()
{
    @Override
    public void
onItemSelected(AdapterView<?> parent,
View view, int position, long id) {
        // Handle when item selected
    }

    @Override
    public void
onNothingSelected(AdapterView<?>
parent) {
        // Do nothing
    }
});
}

```

```

private void updateData() {
    String jenis =
editJenisAy.getSelectedItem().toString();
    String kandangAyam =
editKandang.getSelectedItem().toString();
    String status =
editKondisi.getSelectedItem().toString();
    int jumlahAyam =
Integer.parseInt(editJumlahAyam.getText(

```

```

).toString());

    int day =
datePicker.getDayOfMonth();
    int month = datePicker.getMonth() +
1; // DatePicker month is zero-based
    int year = datePicker.getYear();
    String tanggalMasukTelur = year + "-"
+ month + "-" + day;

    // Update data di Firebase

databaseReference.child(idAyam).child("je
nis").setValue(jenis);

databaseReference.child(idAyam).child("k
andangAyam").setValue(kandangAyam);

databaseReference.child(idAyam).child("st
atus").setValue(status);

databaseReference.child(idAyam).child("ju
mlahAyam").setValue(jumlahAyam);

databaseReference.child(idAyam).child("ta
nggal").setValue(tanggalMasukTelur);

    Toast.makeText(Edit_Ayam.this,
"Data berhasil diupdate",
Toast.LENGTH_SHORT).show();
    finish();
}

```

```

private void hapusData() {

databaseReference.child(idAyam).remove
Value();

    Toast.makeText(Edit_Ayam.this,
"Data berhasil dihapus",
Toast.LENGTH_SHORT).show();

```

```

        finish();
    }
}

```

table_ayam.java

```

package
com.aplikasi.apliaksiternaktelur.admin.aya
m;

import android.graphics.Color;
import android.os.Bundle;
import android.view.Gravity;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;

import com.aplikasi.apliaksiternaktelur.R;
import
com.google.firebase.database.DataSnapsho
t;
import
com.google.firebase.database.DatabaseErr
or;
import
com.google.firebase.database.DatabaseRef
erence;
import
com.google.firebase.database.FirebaseData
base;
import
com.google.firebase.database.ValueEventL
istener;

import java.util.HashMap;
import java.util.Map;

public class table_ayam extends
AppCompatActivity {

```

```

    private DatabaseReference
databaseReference;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_table_ay
am);

        // Inisialisasi Firebase Database
        databaseReference =
FirebaseDatabase.getInstance().getReferen
ce("admin/ayam");

        // Ambil data dan update tabel
        fetchDataFromFirebase();
    }

    private void fetchDataFromFirebase() {

        databaseReference.addValueEventListener
(new ValueEventListener() {
            @Override
            public void
onDataChange(@NonNull DataSnapshot
snapshot) {
                // Struktur data untuk
                menyimpan kategori per tanggal
                Map<String, Map<String,
Integer>> dataByDate = new
HashMap<>();

                for (DataSnapshot dataSnapshot
: snapshot.getChildren()) {
                    String tanggal =
dataSnapshot.child("tanggal").getValue(Str
ing.class);
                    String status =
dataSnapshot.child("status").getValue(Stri
ng.class);
                    int jumlahAyam =
dataSnapshot.child("jumlahAyam").getVal
ue(Integer.class);

```

```

        // Inisialisasi data per tanggal
        jika belum ada
        if
        (!dataByDate.containsKey(tanggal)) {
            dataByDate.put(tanggal,
            new HashMap<String, Integer>() {{
                put("Sehat", 0);
                put("Sakit", 0);
                put("Mati", 0);
                put("Total", 0);
            }});
        }

        // Tambah jumlah ayam
        berdasarkan status
        Map<String, Integer> stats =
        dataByDate.get(tanggal);
        if
        (status.equalsIgnoreCase("Sehat")) {
            stats.put("Sehat",
            stats.get("Sehat") + jumlahAyam);
        } else if
        (status.equalsIgnoreCase("Sakit")) {
            stats.put("Sakit",
            stats.get("Sakit") + jumlahAyam);
        } else if
        (status.equalsIgnoreCase("Mati")) {
            stats.put("Mati",
            stats.get("Mati") + jumlahAyam);
        }
        stats.put("Total",
        stats.get("Total") + jumlahAyam);
    }

    // Update tabel dengan data
    updateTable(dataByDate);
}

@Override
public void
onCancelled(@NonNull DatabaseError
error) {
    Toast.makeText(table_ayam.this,
"Gagal mengambil data: " +
error.getMessage(),
Toast.LENGTH_SHORT).show();

```

```

    }
    });
}

private void updateTable(Map<String,
Map<String, Integer>> dataByDate) {
    TableLayout tableLayout =
    findViewById(R.id.tableLayout);

    // Clear existing rows
    tableLayout.removeAllViews();

    // Add header row
    TableRow headerRow = new
    TableRow(this);
    String[] headers = {"Tanggal",
    "Sakit", "Sehat", "Mati", "Total"};
    for (String header : headers) {
        TextView textView = new
        TextView(this);
        textView.setText(header);
        textView.setPadding(8, 8, 8, 8);

        textView.setTextColor(Color.WHITE);

        textView.setBackgroundColor(Color.DKGRAY);

        textView.setGravity(Gravity.CENTER);

        TableRow.LayoutParams params =
        new TableRow.LayoutParams(
            0, // width: match_parent

            TableRow.LayoutParams.WRAP_CONTENT,
            1.0f // weight
        );

        textView.setLayoutParams(params);
        headerRow.addView(textView);
    }
    tableLayout.addView(headerRow);

    // Add data rows
    for (Map.Entry<String, Map<String,

```

```

Integer>> entry : dataByDate.entrySet() {
    String tanggal = entry.getKey();
    Map<String, Integer> stats =
entry.getValue();

    TableRow row = new
TableRow(this);
    String[] data = {
        tanggal,

String.valueOf(stats.get("Sakit")),

String.valueOf(stats.get("Sehat")),

String.valueOf(stats.get("Mati")),

String.valueOf(stats.get("Total"))
    };

    for (String cellData : data) {
        TextView cell = new
TextView(this);
        cell.setText(cellData);
        cell.setPadding(8, 8, 8, 8);

cell.setGravity(Gravity.CENTER);

        TableRow.LayoutParams params
= new TableRow.LayoutParams(
            0, // width: match_parent

TableRow.LayoutParams.WRAP_CONTE
NT,

            1.0f // weight
        );
        cell.setLayoutParams(params);
        row.addView(cell);
    }

    tableLayout.addView(row);
}
}
}

```

TambahAyam.java

```

package
com.aplikasi.apliaksiternaktelur.admin.aya
m;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;

import android.app.DatePickerDialog;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import
com.aplikasi.apliaksiternaktelur.MainActiv
ity;
import com.aplikasi.apliaksiternaktelur.R;
import
com.aplikasi.apliaksiternaktelur.admin.telur.TambahTelur;
import
com.google.android.gms.tasks.OnFailureL
istener;
import
com.google.android.gms.tasks.OnSuccess
Listener;
import
com.google.firebase.database.DatabaseRef
erence;
import
com.google.firebase.database.FirebaseData
base;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;

```

```

public class TambahAyam extends
AppCompatActivity {

    private EditText editTextJumlahAyam;
    private Spinner spinnerKandang,
spinnerJenisAyam, spinnerKondisi;
    private DatePicker datePicker;
    private Button btnSave;
    private TextView tvDatePicker;
    // Referensi ke Firebase Database
    private DatabaseReference
databaseReference;

    // Array untuk jenis ayam dan kondisi
    private String[] jenisAyamArray;
    private String[] kandangArray;
    private String[] kondisiArray;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_tambah_
ayam);

        // Inisialisasi Firebase Database
        databaseReference =
FirebaseDatabase.getInstance().getReferen
ce().child("admin/ayam");

        // Inisialisasi views
        editTextJumlahAyam =
findViewById(R.id.editTextAmountAy);
        spinnerKandang =
findViewById(R.id.spinnerKandangAy);
        spinnerJenisAyam =
findViewById(R.id.spinnerChickenTypeAy
);
        spinnerKondisi =
findViewById(R.id.editTextKondisi);
        btnSave =
findViewById(R.id.buttonSave);
        tvDatePicker =
findViewById(R.id.tvDatePicker);
        // Ambil array jenis ayam, kandang,

```

```

dan kondisi dari resources
        jenisAyamArray =
getResources().getStringArray(R.array.list
JenisAyam);
        kandangArray =
getResources().getStringArray(R.array.list
Kandang);
        kondisiArray =
getResources().getStringArray(R.array.list
Kondisi);

        // Setup ArrayAdapter untuk Spinner
jenis ayam
        ArrayAdapter<CharSequence>
jenisAyamAdapter =
ArrayAdapter.createFromResource(this,
R.array.listJenisAyam,
android.R.layout.simple_spinner_item);

jenisAyamAdapter.setDropDownViewResou
rce(android.R.layout.simple_spinner_dr
opdown_item);

spinnerJenisAyam.setAdapter(jenisAyamA
dapter);

        // Setup ArrayAdapter untuk Spinner
kandang
        ArrayAdapter<CharSequence>
kandangAdapter =
ArrayAdapter.createFromResource(this,
R.array.listKandang,
android.R.layout.simple_spinner_item);

kandangAdapter.setDropDownViewResou
rce(android.R.layout.simple_spinner_drop
down_item);

spinnerKandang.setAdapter(kandangAdapt
er);

        // Setup ArrayAdapter untuk Spinner
kondisi
        ArrayAdapter<CharSequence>
kondisiAdapter =
ArrayAdapter.createFromResource(this,

```

```

        R.array.listKondisi,
        android.R.layout.simple_spinner_item);

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

spinnerKondisi.setAdapter(kondisiAdapter);

```

```

        // Tampilkan tanggal saat ini di
        TextView saat aplikasi dibuka
        SimpleDateFormat sdf = new
        SimpleDateFormat("dd-MM-yyyy",
        Locale.getDefault());
        String currentDate =
        sdf.format(Calendar.getInstance().getTime());
        tvDatePicker.setText(currentDate);

```

```

        tvDatePicker.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Calendar calendar =
                Calendar.getInstance();
                int year =
                calendar.get(Calendar.YEAR);
                int month =
                calendar.get(Calendar.MONTH);
                int day =
                calendar.get(Calendar.DAY_OF_MONTH);

```

```

                DatePickerDialog
                datePickerDialog = new
                DatePickerDialog(TambahAyam.this,
                new
                DatePickerDialog.OnDateSetListener() {
                    @Override
                    public void
                    onDateSet(DatePicker view, int year, int
                    month, int dayOfMonth) {
                        Calendar selectedDate
                        = Calendar.getInstance();
                        selectedDate.set(year,

```

```

month, dayOfMonth);
                        String date =
                        sdf.format(selectedDate.getTime());

                        tvDatePicker.setText(date);
                    }
                }, year, month, day);
                datePickerDialog.show();
            }
        });

```

```

        // Setup button click listener
        btnSave.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                tambahDataAyam();
            }
        });
    }

```

```

        private void tambahDataAyam() {
            // Ambil nilai dari input user
            String jumlahAyamStr =
            editTextJumlahAyam.getText().toString().trim();
            String kandang =
            spinnerKandang.getSelectedItem().toString();
            String jenisAyam =
            spinnerJenisAyam.getSelectedItem().toString();
            String kondisi =
            spinnerKondisi.getSelectedItem().toString();

            String tanggalMasukAyam =
            tvDatePicker.getText().toString().trim();
            // Validasi apakah ada input yang
            kosong
            if (jumlahAyamStr.isEmpty() ||
            kandang.equals("Pilih") ||
            jenisAyam.equals("Pilih") ||
            kondisi.equals("Pilih")) {
                Toast.makeText(this, "Harap
                lengkapi semua data",
                Toast.LENGTH_SHORT).show();

```

```

        return;
    }

    // Ambil nilai jumlahAyam setelah
    validasi
    int jumlahAyam =
    Integer.parseInt(jumlahAyamStr);

    // Buat objek DataAyam
    String idAyam =
    databaseReference.push().getKey(); //
    Generate unique key
    DataAyam dataAyam = new
    DataAyam(idAyam, jenisAyam,
    jumlahAyam, kandang, kondisi,
    tanggalMasukAyam);

    // Simpan data ke Firebase

    databaseReference.child(idAyam).setValue
    (dataAyam)
        .addOnSuccessListener(new
    OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void
    aVoid) {

    Toast.makeText(TambahAyam.this, "Data
    ayam berhasil disimpan",
    Toast.LENGTH_SHORT).show();
            // Reset form setelah data
    disimpan (opsional)
            resetForm();
            Intent intent = new
    Intent(TambahAyam.this,
    AyamActivity.class);
            startActivity(intent);
            finish();
        }
    })
        .addOnFailureListener(new
    OnFailureListener() {
            @Override
            public void

```

```

onFailure(@NonNull Exception e) {

    Toast.makeText(TambahAyam.this, "Gagal
    menyimpan data: " + e.getMessage(),
    Toast.LENGTH_SHORT).show();
        }
    });
}

private void resetForm() {
    editTextJumlahAyam.setText("");
    spinnerKandang.setSelection(0); //
    Reset Spinner selection
    spinnerJenisAyam.setSelection(0); //
    Reset Spinner selection
    spinnerKondisi.setSelection(0); //
    Reset Spinner selection
    tvDatePicker.setText(""); //
    Mengosongkan TextView
}

public void pressback(View view) {
    Intent intent = new
    Intent(TambahAyam.this,
    AyamActivity.class);
    startActivity(intent);
    finish();
}
}

```

Laporan Ayam

Coba.java

```

package
com.aplikasi.apliksiternaktelur.admin.lap
oran;

```

```

import android.os.Bundle;
import android.util.Log;
import android.widget.Toast;

```

```

import androidx.annotation.NonNull;
import

```

```
androidx.appcompat.app.AppCompatActivity;
```

```
import com.aplikasi.apliaksiternaktelur.R;
import
com.github.mikephil.charting.charts.BarChart;
import
com.github.mikephil.charting.components.XAxis;
import
com.github.mikephil.charting.components.YAxis;
import
com.github.mikephil.charting.data.BarData;
import
com.github.mikephil.charting.data.BarDataSet;
import
com.github.mikephil.charting.data.BarEntry;
import
com.github.mikephil.charting.utils.ColorTemplate;
import
com.google.firebase.database.DataSnapshot;
import
com.google.firebase.database.DatabaseError;
import
com.google.firebase.database.DatabaseReference;
import
com.google.firebase.database.FirebaseDatabase;
import
com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;

public class coba extends
AppCompatActivity {
```

```
    private DatabaseReference
pemasukanRef;
    private DatabaseReference
pengeluaranRef;
    private BarChart barChart;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_coba);

        // Inisialisasi referensi Firebase
        pemasukanRef =
FirebaseDatabase.getInstance().getReference("admin/pemasukan");
        pengeluaranRef =
FirebaseDatabase.getInstance().getReference("admin/pakan");

        // Inisialisasi BarChart dari layout
XML
        barChart =
findViewById(R.id.bar_chart);

        // Ambil data dan tampilkan grafik
        fetchDataAndDisplayChart();
    }

    private void
fetchDataAndDisplayChart() {

        pemasukanRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void
onDataChange(@NonNull DataSnapshot
snapshot) {
                int smallestPemasukan =
Integer.MAX_VALUE;
                int largestPemasukan =
Integer.MIN_VALUE;
                int totalPemasukan = 0;

                for (DataSnapshot dataSnapshot
```

```

: snapshot.getChildren()) {
    int hargaSatuan =
dataSnapshot.child("hargaSatuan").getValu
e(Integer.class);
    totalPemasukan +=
hargaSatuan;

    // Update nilai terkecil dan
terbesar pemasukan
    if (hargaSatuan <
smallestPemasukan) {
        smallestPemasukan =
hargaSatuan;
    }
    if (hargaSatuan >
largestPemasukan) {
        largestPemasukan =
hargaSatuan;
    }
}

// Setelah mendapatkan
totalPemasukan, lanjutkan untuk
menghitung totalPengeluaran

calculateTotalPengeluaran(smallestPemasu
kan, largestPemasukan);
}

@Override
public void
onCancelled(@NonNull DatabaseError
error) {
    // Handle kegagalan ambil data
pemasukan
    Log.e("FirebaseError", "Gagal
mengambil data pemasukan: " +
error.getMessage());
    Toast.makeText(coba.this,
"Gagal mengambil data pemasukan: " +
error.getMessage(),
Toast.LENGTH_SHORT).show();
}
});
}

```

```

private void
calculateTotalPengeluaran(final int
smallestPemasukan, final int
largestPemasukan) {

pengeluaranRef.addValueEventListener(new
ValueEventListener() {
    @Override
    public void
onDataChange(@NonNull DataSnapshot
snapshot) {
        int smallestPengeluaran =
Integer.MAX_VALUE;
        int largestPengeluaran =
Integer.MIN_VALUE;
        int totalPengeluaran = 0;

        for (DataSnapshot dataSnapshot
: snapshot.getChildren()) {
            Integer jumlahPakan =
dataSnapshot.child("jumlahPakan").getVal
ue(Integer.class);
            Integer hargaPerKg =
dataSnapshot.child("hargaPerKg").getValu
e(Integer.class);
            if (jumlahPakan != null &&
hargaPerKg != null) {
                int totalBiayaPakan =
jumlahPakan * hargaPerKg;
                totalPengeluaran +=
totalBiayaPakan;

                // Update nilai terkecil dan
terbesar pengeluaran
                if (totalBiayaPakan <
smallestPengeluaran) {
                    smallestPengeluaran =
totalBiayaPakan;
                }
                if (totalBiayaPakan >
largestPengeluaran) {
                    largestPengeluaran =
totalBiayaPakan;
                }
            }
        }
    }
}
}

```

```

        // Setelah mendapatkan
        totalPengeluaran, tampilkan grafik

        displayBarChart(smallestPemasukan,
        largestPemasukan, smallestPengeluaran,
        largestPengeluaran);
    }

    @Override
    public void
    onCancelled(@NonNull DatabaseError
    error) {
        // Handle kegagalan ambil data
        pengeluaran
        Log.e("FirebaseError", "Gagal
        mengambil data pengeluaran: " +
        error.getMessage());
        Toast.makeText(coba.this,
        "Gagal mengambil data pengeluaran: " +
        error.getMessage(),
        Toast.LENGTH_SHORT).show();
    }
    });
}

private void displayBarChart(int
smallestPemasukan, int largestPemasukan,
int smallestPengeluaran, int
largestPengeluaran) {
    ArrayList<BarEntry> entries = new
    ArrayList<>();
    entries.add(new BarEntry(0,
    smallestPemasukan, "Pemasukan
    Terkecil"));
    entries.add(new BarEntry(1,
    largestPemasukan, "Pemasukan
    Terbesar"));
    entries.add(new BarEntry(3,
    smallestPengeluaran, "Pengeluaran
    Terkecil"));
    entries.add(new BarEntry(4,
    largestPengeluaran, "Pengeluaran
    Terbesar"));

    BarDataSet barDataSet = new

```

```

BarDataSet(entries, "Pemasukan vs
Pengeluaran");

barDataSet.setColors(ColorTemplate.MAT
ERIAL_COLORS);

    BarData barData = new
    BarData(barDataSet);
    barChart.setData(barData);

    // Customisasi grafik

    barChart.getDescription().setEnabled(false
    ); // Sembunyikan deskripsi

    barChart.setDrawValueAboveBar(true); //
    Tampilkan nilai di atas bar
    barChart.setFitBars(true); // Atur
    lebar bar agar sesuai dengan data
    barChart.setExtraOffsets(5, 10, 5, 5);
    // Atur padding tambahan di sekitar grafik

    // Atur sumbu X
    XAxis xAxis = barChart.getXAxis();
    xAxis.setValueFormatter(new
    com.github.mikephil.charting.formatter.In
    dexAxisValueFormatter(new
    String[]{"PSk", "PSb", "PLk", "PLb"}));

    xAxis.setPosition(XAxis.XAxisPosition.B
    OTTOM);
    xAxis.setGranularity(1f);
    xAxis.setCenterAxisLabels(true);

    // Atur sumbu Y
    YAxis yAxisRight =
    barChart.getAxisRight();
    yAxisRight.setEnabled(false);

    // Atur legenda

    barChart.getLegend().setWordWrapEnable
    d(true);

    barChart.animateXY(1000, 1000); //
    Animasi ketika grafik muncul

```

```

        barChart.invalidate(); // Refresh
    grafik
    }
}

```

StatikActivity.java

```

package
com.aplikasi.apliaksiternaktelur.admin.lap
oran;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import
com.aplikasi.apliaksiternaktelur.MainActiv
ity;
import com.aplikasi.apliaksiternaktelur.R;
import
com.aplikasi.apliaksiternaktelur.admin.Sett
ing;
import
com.github.mikephil.charting.charts.BarC
hart;
import
com.github.mikephil.charting.components.
XAxis;
import
com.github.mikephil.charting.components.
YAxis;
import
com.github.mikephil.charting.data.BarDat
a;
import
com.github.mikephil.charting.data.BarDat
aSet;

```

```

import
com.github.mikephil.charting.data.BarEntr
y;
import
com.github.mikephil.charting.formatter.In
dexAxisValueFormatter;
import
com.github.mikephil.charting.utils.ColorT
emplate;
import
com.google.firebase.database.DataSnapsho
t;
import
com.google.firebase.database.DatabaseErr
or;
import
com.google.firebase.database.DatabaseRef
erence;
import
com.google.firebase.database.FirebaseData
base;
import
com.google.firebase.database.ValueEventL
istener;

```

```

import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.Locale;

```

```

public class StatistikActivity extends
AppCompatActivity {

```

```

    private TextView tvTotalPemasukan;
    private TextView tvTotalPengeluaran;
    private BarChart barChart, barChart2;
    private DatabaseReference
pemasukanRef;
    private DatabaseReference
pengeluaranRef;
    private long totalPemasukanValue = 0;
    private long totalPengeluaranValue = 0;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```
setContentView(R.layout.activity_statistik
2);
```

```
    // Inisialisasi TextView dan BarChart
    tvTotalPemasukan =
    findViewById(R.id.tvTotalPemasukan);
    tvTotalPengeluaran =
    findViewById(R.id.tvTotalPengeluaran);
    barChart =
    findViewById(R.id.barchart);
    barChart2 =
    findViewById(R.id.chart2);
```

```
    // Referensi ke Firebase Database
    pemasukanRef =
    FirebaseDatabase.getInstance().getReferen
    ce("admin/pemasukan");
    pengeluaranRef =
    FirebaseDatabase.getInstance().getReferen
    ce("admin/pakan");
```

```
    ImageButton btnstatistik =
    findViewById(R.id.btnstatistik);
    ImageButton btnberanda =
    findViewById(R.id.btnberanda);
    ImageButton btnakun =
    findViewById(R.id.btnakun);
```

```
    btnstatistik.setBackgroundResource(R.dra
wable.ic_chart_aktif);
```

```
    btnberanda.setBackgroundResource(R.dra
wable.ic_home);
```

```
    btnakun.setBackgroundResource(R.drawa
ble.ic_settings);
```

```
    // Mengambil data dari Firebase
    ambilDataPemasukan();
    ambilDataPengeluaran();
    fetchDataAndDisplayChart();
    fetchDataAndDisplayChart2();
}
```

```
    public void gotoberanda(View view) {
        Intent intent = new
        Intent(StatistikActivity.this,
        MainActivity.class);
        startActivity(intent);
        finish();
    }
```

```
    public void gotopengaturan(View view)
    {
        Intent intent = new
        Intent(StatistikActivity.this, Setting.class);
        startActivity(intent);
        finish();
    }
```

```
    public void gotostatistik(View view) {
        Intent intent = new
        Intent(StatistikActivity.this,
        StatistikActivity.class);
        startActivity(intent);
        finish();
    }
```

```
    private void
    fetchDataAndDisplayChart() {
```

```
        pemasukanRef.addValueEventListener(ne
w ValueEventListener() {
```

```
            @Override
```

```
            public void
            onDataChange(@NonNull DataSnapshot
            snapshot) {
```

```
                int totalPemasukan = 0;
                for (DataSnapshot dataSnapshot
                : snapshot.getChildren()) {
                    Integer hargaSatuan =
                    dataSnapshot.child("totalHarga").getValue
                    (Integer.class);
```

```
                    if (hargaSatuan != null) {
                        totalPemasukan +=
                        hargaSatuan;
                    }
                }
            }
```

```
        int finalTotalPemasukan =
```

```

totalPemasukan;

pengeluaranRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void
onDataChange(@NonNull DataSnapshot
snapshot) {
        int totalPengeluaran = 0;
        for (DataSnapshot
dataSnapshot : snapshot.getChildren()) {
            Integer jumlahPakan =
dataSnapshot.child("jumlahPakan").getVal
ue(Integer.class);
            Integer hargaPerKg =
dataSnapshot.child("hargaPerKg").getValu
e(Integer.class);
            if (jumlahPakan != null
&& hargaPerKg != null) {
                totalPengeluaran +=
hargaPerKg * jumlahPakan;
            }
        }

displayBarChart(finalTotalPemasukan,
totalPengeluaran);
    }

    @Override
    public void
onCancelled(@NonNull DatabaseError
error) {
        Log.e("FirebaseError",
"Gagal mengambil data pengeluaran: " +
error.getMessage());

        Toast.makeText(StatistikActivity.this,
"Gagal mengambil data pengeluaran: " +
error.getMessage(),
Toast.LENGTH_SHORT).show();
    }
});
}

@Override

```

```

    public void
onCancelled(@NonNull DatabaseError
error) {
        Log.e("FirebaseError", "Gagal
mengambil data pemasukan: " +
error.getMessage());

        Toast.makeText(StatistikActivity.this,
"Gagal mengambil data pemasukan: " +
error.getMessage(),
Toast.LENGTH_SHORT).show();
    }
});
}

```

```

    private void displayBarChart(int
totalPemasukan, int totalPengeluaran) {
        ArrayList<BarEntry> entries = new
ArrayList<>();
        entries.add(new BarEntry(0,
totalPemasukan, "Pemasukan"));
        entries.add(new BarEntry(1,
totalPengeluaran, "Pengeluaran"));
    }

```

```

        BarDataSet barDataSet = new
BarDataSet(entries, "Pemasukan vs
Pengeluaran");
        barDataSet.setColors(new
int[]{R.color.bar_color_2,
R.color.bar_color_4}, this);
    }

```

```

        BarData barData = new
BarData(barDataSet);
        barChart.setData(barData);
    }

```

```

        barChart.getDescription().setEnabled(false
);
    }

```

```

        barChart.setDrawValueAboveBar(true);
        barChart.setFitBars(true);
        barChart.setExtraOffsets(5, 10, 5, 5);
    }

```

```

        barChart.getXAxis().setValueFormatter(new
ValueFormatter() {
            @Override
            public String formatValue(float value) {
                return String.format("%.2f", value);
            }
        });
    }
}

```

```

com.github.mikephil.charting.formatter.ValueFormatter() {
    @Override
    public String
    getFormattedValue(float value) {
        if (value == 0) return
        "Pemasukan";
        else if (value == 1) return
        "Pengeluaran";
        else return "";
    }
});

```

```

barChart.getLegend().setWordWrapEnabled(true);

```

```

        barChart.animateXY(1000, 1000);
        barChart.invalidate();
    }

```

```

    private void
    fetchDataAndDisplayChart2() {

        pemasukanRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void
            onDataChange(@NonNull DataSnapshot snapshot) {
                int smallestPemasukan =
                Integer.MAX_VALUE;
                int largestPemasukan =
                Integer.MIN_VALUE;
                int totalPemasukan = 0;

                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
                    Integer hargaSatuan =
                    dataSnapshot.child("totalHarga").getValue
                    (Integer.class);
                    if (hargaSatuan != null) {
                        totalPemasukan +=
                        hargaSatuan;

                        if (hargaSatuan <

```

```

                        smallestPemasukan) {
                            smallestPemasukan =
                            hargaSatuan;
                        }
                        if (hargaSatuan >
                        largestPemasukan) {
                            largestPemasukan =
                            hargaSatuan;
                        }
                    }
                }
            }
        }
    }

```

```

        calculateTotalPengeluaran2(smallestPemasukan, largestPemasukan);
    }

```

```

        @Override
        public void
        onCancelled(@NonNull DatabaseError error) {
            Log.e("FirebaseError", "Gagal
            mengambil data pemasukan: " +
            error.getMessage());

```

```

            Toast.makeText(StatistikActivity.this,
            "Gagal mengambil data pemasukan: " +
            error.getMessage(),
            Toast.LENGTH_SHORT).show();
        }
    });
}

```

```

    private void
    calculateTotalPengeluaran2(final int
    smallestPemasukan, final int
    largestPemasukan) {

```

```

        pengeluaranRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void
            onDataChange(@NonNull DataSnapshot snapshot) {
                int smallestPengeluaran =
                Integer.MAX_VALUE;

```

```

        int largestPengeluaran =
Integer.MIN_VALUE;
        int totalPengeluaran = 0;

        for (DataSnapshot dataSnapshot
: snapshot.getChildren()) {
            Integer jumlahPakan =
dataSnapshot.child("jumlahPakan").getVal
ue(Integer.class);
            Integer hargaPerKg =
dataSnapshot.child("hargaPerKg").getValu
e(Integer.class);
            if (jumlahPakan != null &&
hargaPerKg != null) {
                int totalBiayaPakan =
jumlahPakan * hargaPerKg;
                totalPengeluaran +=
totalBiayaPakan;

                if (totalBiayaPakan <
smallestPengeluaran) {
                    smallestPengeluaran =
totalBiayaPakan;
                }
                if (totalBiayaPakan >
largestPengeluaran) {
                    largestPengeluaran =
totalBiayaPakan;
                }
            }
        }

displayBarChart2(smallestPemasukan,
largestPemasukan, smallestPengeluaran,
largestPengeluaran);
    }

    @Override
    public void
onCancelled(@NonNull DatabaseError
error) {
        Log.e("FirebaseError", "Gagal
mengambil data pengeluaran: " +
error.getMessage());

```

```

Toast.makeText(StatistikActivity.this,
"Gagal mengambil data pengeluaran: " +
error.getMessage(),
Toast.LENGTH_SHORT).show();
    }
});
}

private void displayBarChart2(int
smallestPemasukan, int largestPemasukan,
int smallestPengeluaran, int
largestPengeluaran) {
    ArrayList<BarEntry> entries = new
ArrayList<>();
    entries.add(new BarEntry(0,
smallestPemasukan, "Pemasukan
Terkecil"));
    entries.add(new BarEntry(1,
largestPemasukan, "Pemasukan
Terbesar"));
    entries.add(new BarEntry(3,
smallestPengeluaran, "Pengeluaran
Terkecil"));
    entries.add(new BarEntry(4,
largestPengeluaran, "Pengeluaran
Terbesar"));

    BarDataSet barDataSet = new
BarDataSet(entries, "Pemasukan vs
Pengeluaran");
    barDataSet.setColors(new
int[]{R.color.bar_color_1,
R.color.bar_color_2, R.color.bar_color_3,
R.color.bar_color_4}, this);

    BarData barData = new
BarData(barDataSet);
    barChart2.setData(barData);

    barChart2.getDescription().setEnabled(fals
e);

    barChart2.setDrawValueAboveBar(true);
    barChart2.setFitBars(true);
    barChart2.setExtraOffsets(5, 10, 5,

```

```

5);

XAxis xAxis =
barChart2.getXAxis();
xAxis.setValueFormatter(new
IndexAxisValueFormatter(new
String[]{"PMk", "PMb", "PLk", "PLb"}));

xAxis.setPosition(XAxis.XAxisPosition.BOTTOM);
xAxis.setGranularity(1f);
xAxis.setCenterAxisLabels(true);

YAxis yAxisRight =
barChart2.getAxisRight();
yAxisRight.setEnabled(false);

barChart2.getLegend().setWordWrapEnabled(true);

barChart2.animateXY(1000, 1000);
barChart2.invalidate();
}

private void ambilDataPemasukan() {
pemasukanRef.addValueEventListener(new ValueEventListener() {
@Override
public void
onDataChange(@NonNull DataSnapshot dataSnapshot) {
totalPemasukanValue = 0;
for (DataSnapshot snapshot :
dataSnapshot.getChildren()) {
Integer totalHarga =
snapshot.child("totalHarga").getValue(Integer.class);
if (totalHarga != null) {
totalPemasukanValue +=
totalHarga;
}
}
Locale localeID = new
Locale("in", "ID");

```

```

NumberFormat numberFormat =
NumberFormat.getCurrencyInstance(localeID);

String formattedPemasukan =
numberFormat.format(totalPemasukanValue);

tvTotalPemasukan.setText(formattedPemasukan);
}

@Override
public void
onCancelled(@NonNull DatabaseError databaseError) {
Log.e("FirebaseError", "Gagal
mengambil data pemasukan: " +
databaseError.getMessage());
}
});
}

private void ambilDataPengeluaran() {
pengeluaranRef.addValueEventListener(new ValueEventListener() {
@Override
public void
onDataChange(@NonNull DataSnapshot dataSnapshot) {
totalPengeluaranValue = 0;
for (DataSnapshot snapshot :
dataSnapshot.getChildren()) {
Integer jumlahPakan =
snapshot.child("jumlahPakan").getValue(Integer.class);
Integer hargaPerKg =
snapshot.child("hargaPerKg").getValue(Integer.class);
if (jumlahPakan != null &&
hargaPerKg != null) {
totalPengeluaranValue +=
jumlahPakan * hargaPerKg;
}
}
Locale localeID = new

```

```

Locale("in", "ID");
        NumberFormat numberFormat =
NumberFormat.getCurrencyInstance(local
eID);
        String formattedPengeluaran =
numberFormat.format(totalPengeluaranVal
ue);

tvTotalPengeluaran.setText(formattedPeng
eluaran);
    }

    @Override
    public void
onCancelled(@NonNull DatabaseError
databaseError) {
        Log.e("FirebaseError", "Gagal
mengambil data pengeluaran: " +
databaseError.getMessage());
    }
    });
}
}
}

```

StatikActivitybkup.java

```

package
com.aplikasi.apliaksiternaktelur.admin.lap
oran;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;
import android.widget.TextView;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;

import
com.aplikasi.apliaksiternaktelur.MainActiv
ity;
import com.aplikasi.apliaksiternaktelur.R;
import

```

```

com.aplikasi.apliaksiternaktelur.admin.Sett
ing;
import
com.github.mikephil.charting.charts.BarC
hart;
import
com.github.mikephil.charting.components.
Description;
import
com.github.mikephil.charting.components.
XAxis;
import
com.github.mikephil.charting.components.
YAxis;
import
com.github.mikephil.charting.data.BarDat
a;
import
com.github.mikephil.charting.data.BarDat
aSet;
import
com.github.mikephil.charting.data.BarEntr
y;
import
com.github.mikephil.charting.formatter.In
dexAxisValueFormatter;
import
com.google.firebase.database.DataSnapsho
t;
import
com.google.firebase.database.DatabaseErr
or;
import
com.google.firebase.database.DatabaseRef
erence;
import
com.google.firebase.database.FirebaseData
base;
import
com.google.firebase.database.ValueEventL
istener;

import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

```

```
public class StatistikActivitybkup extends
AppCompatActivity {
```

```
    private TextView tvTotalPemasukan;
    private TextView tvTotalPengeluaran;
    private BarChart barChart;
    private DatabaseReference
pemasukanRef;
    private DatabaseReference
pengeluaranRef;
    private long totalPemasukanValue = 0;
    private long totalPengeluaranValue = 0;
```

```
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_statistik
2);
```

```
        // Inisialisasi TextView dan BarChart
        tvTotalPemasukan =
findViewById(R.id.tvTotalPemasukan);
        tvTotalPengeluaran =
findViewById(R.id.tvTotalPengeluaran);
        barChart =
findViewById(R.id.barchart);
```

```
        // Referensi ke Firebase Database
        pemasukanRef =
FirebaseDatabase.getInstance().getReferen
ce("admin/pemasukan");
        pengeluaranRef =
FirebaseDatabase.getInstance().getReferen
ce("admin/pakan");
```

```
        ImageButton btnstatistik =
findViewById(R.id.btnstatistik);
        ImageButton btnberanda =
findViewById(R.id.btnberanda);
        ImageButton btnakun =
```

```
findViewById(R.id.btnakun);
```

```
        btnstatistik.setBackgroundResource(R.dra
wable.ic_chart_aktif);
```

```
        btnberanda.setBackgroundResource(R.dra
wable.ic_home);
```

```
        btnakun.setBackgroundResource(R.dra
wable.ic_settings);
```

```
        // Mengambil data dari Firebase
        ambilDataPemasukan();
        ambilDataPengeluaran();
        ambilDataGrafik();
    }
    public void gotoberanda(View view) {
        Intent intent = new
Intent(StatistikActivitybkup.this,
MainActivity.class);
        startActivity(intent);
        finish();
    }
```

```
    public void gotopengaturan(View view)
{
        Intent intent = new
Intent(StatistikActivitybkup.this,
Setting.class);
        startActivity(intent);
        finish();
    }
```

```
    public void gotostatistik(View view) {
        Intent intent = new
Intent(StatistikActivitybkup.this,
StatistikActivitybkup.class);
        startActivity(intent);
        finish();
    }
```

```

private void ambilDataPemasukan() {

pemasukanRef.addListenerForSingleValue
Event(new ValueEventListener() {
    @Override
    public void
onDataChange(@NonNull DataSnapshot
dataSnapshot) {
        long totalPemasukan = 0;

        for (DataSnapshot snapshot :
dataSnapshot.getChildren()) {
            Long jumlahPemasukan =
snapshot.child("totalHarga").getValue(Lon
g.class);

            if (jumlahPemasukan != null)
{
                totalPemasukan +=
jumlahPemasukan;
            }
        }

        // Mengubah totalPemasukan
menjadi format rupiah
        NumberFormat formatter =
NumberFormat.getCurrencyInstance(new
Locale("id", "ID"));
        String formattedTotal =
formatter.format(totalPemasukan);

        // Menambahkan titik di depan
simbol mata uang "Rp"
        String currencySymbol =
formatter.getCurrency().getSymbol(new
Locale("id", "ID"));
        formattedTotal = "Rp " +
formattedTotal.replace(currencySymbol,
""");

        // Menampilkan jumlah
pemasukan pada TextView

tvTotalPemasukan.setText(formattedTotal)
;

```

```

// Simpan nilai total pemasukan
totalPemasukanValue =
totalPemasukan;
    }

    @Override
    public void
onCancelled(@NonNull DatabaseError
error) {
        // Handle failed database read
    }
});
}

private void ambilDataPengeluaran() {

pengeluaranRef.addListenerForSingleValu
eEvent(new ValueEventListener() {
    @Override
    public void
onDataChange(@NonNull DataSnapshot
dataSnapshot) {
        long totalPengeluaran = 0;

        for (DataSnapshot snapshot :
dataSnapshot.getChildren()) {
            Long jumlahPengeluaran =
snapshot.child("hargaPerKg").getValue(Lo
ng.class);

            if (jumlahPengeluaran != null)
{
                totalPengeluaran +=
jumlahPengeluaran;
            }
        }

        // Mengubah totalPengeluaran
menjadi format rupiah
        NumberFormat formatter =
NumberFormat.getCurrencyInstance(new
Locale("id", "ID"));
        String formattedTotal =
formatter.format(totalPengeluaran);

        // Menambahkan titik di depan
simbol mata uang "Rp"

```

```

        String currencySymbol =
formatter.getCurrency().getSymbol(new
Locale("id", "ID"));
        formattedTotal = "Rp " +
formattedTotal.replace(currencySymbol,
""");

        // Menampilkan jumlah
pengeluaran pada TextView

tvTotalPengeluaran.setText(formattedTotal
);

        // Simpan nilai total pengeluaran
totalPengeluaranValue =
totalPengeluaran;
    }

    @Override
    public void
onCancelled(@NonNull DatabaseError
error) {
        // Handle failed database read
    }
    });
}

private void ambilDataGrafik() {

pemasukanRef.addListenerForSingleValue
Event(new ValueEventListener() {
    @Override
    public void
onDataChange(@NonNull DataSnapshot
dataSnapshot) {
        List<BarEntry>
pemasukanEntries = new ArrayList<>();
        List<BarEntry>
pengeluaranEntries = new ArrayList<>();
        List<String> labels = new
ArrayList<>();

        int i = 0;
        for (DataSnapshot snapshot :
dataSnapshot.getChildren()) {
            Long jumlahPemasukan =

```

```

snapshot.child("totalHarga").getValue(Long.class);
            if (jumlahPemasukan != null)
            {
                pemasukanEntries.add(new
BarEntry(i, jumlahPemasukan));
            }
            labels.add("Data " + (i+1)); //
Add label for each entry
            i++;
        }

        // Tambahkan nilai total
pemasukan ke grafik
        pemasukanEntries.add(new
BarEntry(i, totalPemasukanValue));

        BarDataSet pemasukanDataSet
= new BarDataSet(pemasukanEntries,
"Pemasukan");

pemasukanDataSet.setColor(getResources(
).getColor(R.color.blue)); // Set color to
blue

ambilDataGrafikPengeluaran(pemasukanD
ataSet, labels);
    }

    @Override
    public void
onCancelled(@NonNull DatabaseError
error) {
        // Handle failed database read
    }
    });
}

private void
ambilDataGrafikPengeluaran(final
BarDataSet pemasukanDataSet, final
List<String> labels) {

pengeluaranRef.addListenerForSingleValu
eEvent(new ValueEventListener() {

```

```

@Override
public void
onDataChange(@NonNull DataSnapshot
dataSnapshot) {
    List<BarEntry>
    pengeluaranEntries = new ArrayList<>();

    int i = 0;
    for (DataSnapshot snapshot :
dataSnapshot.getChildren()) {
        Long jumlahPengeluaran =
snapshot.child("hargaPerKg").getValue(Long.class);
        if (jumlahPengeluaran != null)
        {

            pengeluaranEntries.add(new BarEntry(i++,
jumlahPengeluaran));
        }

        // Tambahkan nilai total
        pengeluaran ke grafik
        pengeluaranEntries.add(new
BarEntry(i, totalPengeluaranValue));

        BarDataSet pengeluaranDataSet
= new BarDataSet(pengeluaranEntries,
"Pengeluaran");

        pengeluaranDataSet.setColor(getResources().getColor(R.color.red)); // Set color to red

        BarData barData = new
BarData(pemasukanDataSet,
pengeluaranDataSet);
        barData.setBarWidth(0.45f); //
set custom bar width
        barChart.setData(barData);

        // Set up X axis
        XAxis xAxis =
barChart.getXAxis();
        xAxis.setValueFormatter(new
IndexAxisValueFormatter(labels));

```

```

xAxis.setPosition(XAxis.XAxisPosition.BOTTOM);
        xAxis.setGranularity(1f);
        xAxis.setGranularityEnabled(true);

        // Set up Y axis
        YAxis leftAxis =
barChart.getAxisLeft();
        YAxis rightAxis =
barChart.getAxisRight();
        leftAxis.setGranularity(1f);
        rightAxis.setGranularity(1f);

        // Set description
        Description description = new
Description();
        description.setText(" ");

        barChart.setDescription(description);

        // Customize the chart
        appearance
        barChart.groupBars(0f, 0.08f,
0.02f); // group bars with a custom spacing
        barChart.setFitBars(true); //
make the x-axis fit exactly all bars
        barChart.invalidate(); // Refresh
        chart

        // Customize the chart
        appearance
        barChart.groupBars(0f, 0.08f,
0.02f); // group bars with a custom spacing
        barChart.setFitBars(true); //
make the x-axis fit exactly all bars
        barChart.invalidate(); // Refresh
        chart
    }

    @Override
    public void
onCancelled(@NonNull DatabaseError
error) {
        // Handle failed database read
    }

```

```

    });

}

}

```

TableLabaActivity.java

```

package
com.aplikasi.apliaksiternaktelur.admin.lap
oran;

import android.app.DatePickerDialog;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;

import com.aplikasi.apliaksiternaktelur.R;
import
com.github.mikephil.charting.charts.BarC
hart;
import
com.github.mikephil.charting.data.BarDat
a;
import
com.github.mikephil.charting.data.BarDat
aSet;
import
com.github.mikephil.charting.data.BarEntr
y;
import
com.github.mikephil.charting.formatter.In
dexAxisValueFormatter;
import
com.github.mikephil.charting.components.
XAxis;

```

```

import
com.github.mikephil.charting.interfaces.da
tasets.IBarDataSet;
import
com.google.firebase.database.DataSnapsho
t;
import
com.google.firebase.database.DatabaseErr
or;
import
com.google.firebase.database.DatabaseRef
erence;
import
com.google.firebase.database.FirebaseData
base;
import
com.google.firebase.database.ValueEventL
istener;

```

```

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.Locale;

```

```

public class TableLabaActivity extends
AppCompatActivity {

```

```

    private TableLayout tableLayout;
    private TextView startDateTextView,
endDateTextView;
    private Button filterButton;
    private BarChart barChart;
    private SimpleDateFormat sdf;
    private Calendar calendar;

```

```

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

    setContentView(R.layout.activity_table_la
ba);

```

```

        tableLayout =

```

```

findViewById(R.id.tableLayout);
    startDateTextView =
findViewById(R.id.startDate);
    endDateTextView =
findViewById(R.id.endDate);
    filterButton =
findViewById(R.id.filterButton);
    barChart =
findViewById(R.id.barChart);
    sdf = new SimpleDateFormat("dd-
MM-yyyy", Locale.getDefault()); //
Update to match Firebase date format
    calendar = Calendar.getInstance();

    // Set today's date as default for start
and end date
    String todayDate =
sdf.format(calendar.getTime());

startDateTextView.setText(todayDate);
    endDateTextView.setText(todayDate);

startDateTextView.setOnClickListener(ne
w View.OnClickListener() {
    @Override
    public void onClick(View v) {

showDatePickerDialog(startDateTextView
);
    }
});

endDateTextView.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {

showDatePickerDialog(endDateTextView)
;
    }
});

filterButton.setOnClickListener(new
View.OnClickListener() {

```

```

@Override
    public void onClick(View v) {
        String startDate =
startDateTextView.getText().toString();
        String endDate =
endDateTextView.getText().toString();
        loadLabaData(startDate,
endDate);
    }
});

// Load initial data without filter
loadLabaData(null, null);
}

private void
showDatePickerDialog(final TextView
dateTextView) {
    final Calendar calendar =
Calendar.getInstance();
    DatePickerDialog datePickerDialog =
new DatePickerDialog(this, new
DatePickerDialog.OnDateSetListener() {
        @Override
        public void onDateSet(DatePicker
view, int year, int monthOfYear, int
dayOfMonth) {
            calendar.set(Calendar.YEAR,
year);
            calendar.set(Calendar.MONTH,
monthOfYear);

calendar.set(Calendar.DAY_OF_MONTH,
dayOfMonth);

dateTextView.setText(sdf.format(calendar.
getTime()));
        }
    }, calendar.get(Calendar.YEAR),
calendar.get(Calendar.MONTH),
calendar.get(Calendar.DAY_OF_MONTH));
    datePickerDialog.show();
}

private void loadLabaData(String

```

```

startDate, String endDate) {
    DatabaseReference adminRef =
    FirebaseDatabase.getInstance().getReference("admin");

    adminRef.addValueEventListener(new
    ValueEventListener() {
        @Override
        public void
        onDataChange(@NonNull DataSnapshot
        snapshot) {
            tableLayout.removeAllViews();
            barChart.clear();

            // Add table header
            TableRow header = new
            TableRow(TableLabaActivity.this);

            header.addView(createTextView("Tanggal
            "));

            header.addView(createTextView("Pemasu
            kan"));

            header.addView(createTextView("Pengelu
            aran"));

            header.addView(createTextView("Laba"));
            tableLayout.addView(header);

            ArrayList<BarEntry>
            barEntriesPemasukan = new
            ArrayList<>();
            ArrayList<BarEntry>
            barEntriesPengeluaran = new
            ArrayList<>();
            ArrayList<String> xLabels =
            new ArrayList<>();
            int index = 0;

            try {
                Date start = startDate != null ?
                sdf.parse(startDate) : null;
                Date end = endDate != null ?
                sdf.parse(endDate) : null;

```

```

int totalPemasukan = 0;
int totalPengeluaranPakan = 0;
int totalPengeluaranObat = 0;

// Process pemasukan data
if
(snapshot.child("pemasukan").exists()) {
    for (DataSnapshot
    pemasukan :
    snapshot.child("pemasukan").getChildren(
    )) {
        String tanggalPenjualan
        =
        pemasukan.child("tanggalPenjualan").getV
        alue(String.class);
        Date tanggal =
        sdf.parse(tanggalPenjualan);

        if (start != null && end
        != null && (tanggal.before(start) ||
        tanggal.after(end))) {
            continue;
        }

        int totalHarga =
        pemasukan.child("totalHarga").getValue(I
        nteger.class);
        totalPemasukan +=
        totalHarga;

        // Add data for BarChart

        barEntriesPemasukan.add(new
        BarEntry(index, totalHarga));

        xLabels.add(tanggalPenjualan);
        index++;
    }
}

// Process obat data
if
(snapshot.child("obat").exists()) {
    for (DataSnapshot obat :
    snapshot.child("obat").getChildren()) {
        String tanggalObat =

```

```
obat.child("tanggal").getValue(String.class);
```

```
        Date tanggal =  
sdf.parse(tanggalObat);
```

```
        if (start != null && end  
!= null && (tanggal.before(start) ||  
tanggal.after(end))) {  
            continue;  
        }
```

```
        int hargaObat =  
obat.child("hargaObat").getValue(Integer.c  
lass);
```

```
        int jumlahObat =  
obat.child("jumlahObat").getValue(Integer.  
class);
```

```
        totalPengeluaranObat +=  
hargaObat * jumlahObat;
```

```
        // Add data for BarChart
```

```
barEntriesPengeluaran.add(new  
BarEntry(index, hargaObat *  
jumlahObat));
```

```
xLabels.add(tanggalObat);  
        index++;  
    }  
}
```

```
        // Process pakan data  
        if  
(snapshot.child("pakan").exists()) {  
            for (DataSnapshot pakan :  
snapshot.child("pakan").getChildren()) {  
                String tanggalPemberian  
=  
pakan.child("tanggalPemberian").getValue  
(String.class);
```

```
                Date tanggal =  
sdf.parse(tanggalPemberian);
```

```
                if (start != null && end  
!= null && (tanggal.before(start) ||  
tanggal.after(end))) {
```

```
                    continue;  
                }
```

```
                int hargaPerKg =  
pakan.child("hargaPerKg").getValue(Integ  
er.class);
```

```
                int jumlahPakan =  
pakan.child("jumlahPakan").getValue(Inte  
ger.class);
```

```
                totalPengeluaranPakan  
+= hargaPerKg * jumlahPakan;
```

```
                // Add data for BarChart
```

```
barEntriesPengeluaran.add(new  
BarEntry(index, hargaPerKg *  
jumlahPakan));
```

```
xLabels.add(tanggalPemberian);  
        index++;  
    }  
}
```

```
        int totalPengeluaran =  
totalPengeluaranPakan +  
totalPengeluaranObat;  
        int laba = totalPemasukan -  
totalPengeluaran;
```

```
        TableRow row = new  
TableRow(TableLabaActivity.this);
```

```
row.addView(createTextView("Total"));
```

```
row.addView(createTextView(String.value  
Of(totalPemasukan)));
```

```
row.addView(createTextView(String.value  
Of(totalPengeluaran)));
```

```
row.addView(createTextView(String.value  
Of(laba)));
```

```
        tableLayout.addView(row);
```

```
        // Prepare data for BarChart  
BarDataSet
```

```

barDataSetPemasukan = new
BarDataSet(barEntriesPemasukan,
"Pemasukan");

barDataSetPemasukan.setColor(getResour
ces().getColor(R.color.colorPemasukan));
// Set color for Pemasukan
BarDataSet
barDataSetPengeluaran = new
BarDataSet(barEntriesPengeluaran,
"Pengeluaran");

barDataSetPengeluaran.setColor(getResou
rces().getColor(R.color.colorPengeluaran)
); // Set color for Pengeluaran

```

```

ArrayList<IBarDataSet>
dataSets = new ArrayList<>();

dataSets.add(barDataSetPemasukan);

dataSets.add(barDataSetPengeluaran);

```

```

BarData barData = new
BarData(dataSets);
barChart.setData(barData);
barChart.invalidate(); //
Refresh chart

```

```

// Set X Axis labels
XAxis xAxis =
barChart.getXAxis();
xAxis.setValueFormatter(new
IndexAxisValueFormatter(xLabels));

xAxis.setPosition(XAxis.XAxisPosition.B
OTTOM);
xAxis.setGranularity(1f); //
Show each label

```

```

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

```

```

@Override

```

```

public void
onCancelled(@NonNull DatabaseError
error) {
    // Handle possible errors.
}
});
}

```

```

private TextView createTextView(String
text) {
    TextView textView = new
TextView(this);
textView.setText(text);
textView.setPadding(8, 8, 8, 8);

textView.setBackgroundResource(R.drawa
ble.cell_border); // Apply cell border
drawable
return textView;
}
}

```

TablePemasukanActivity.java

```

package
com.aplikasi.apliaksiternaktelur.admin.lap
oran;

```

```

import android.app.DatePickerDialog;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;

```

```

import com.aplikasi.apliaksiternaktelur.R;

```

```

import
com.github.mikephil.charting.data.BarEntr
y;

```

```

import
com.github.mikephil.charting.data.BarData;
import
com.github.mikephil.charting.data.BarDataSet;
import
com.github.mikephil.charting.charts.BarChart;
import
com.github.mikephil.charting.components.XAxis;
import
com.github.mikephil.charting.formatter.IndexAxisValueFormatter;
import
com.google.firebase.database.DataSnapshot;
import
com.google.firebase.database.DatabaseError;
import
com.google.firebase.database.DatabaseReference;
import
com.google.firebase.database.FirebaseDatabase;
import
com.google.firebase.database.ValueEventListener;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;
import java.util.Locale;
import java.util.Map;

public class TablePemasukanActivity
extends AppCompatActivity {

    private TableLayout tableLayout;
    private TextView startDateTextView,
endDateTextView;
    private Button filterButton;
    private BarChart barChart;

```

```

    private SimpleDateFormat sdf;
    private Calendar calendar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_table_pemasukan);

        tableLayout =
findViewById(R.id.tableLayout);
        startDateTextView =
findViewById(R.id.startDate);
        endDateTextView =
findViewById(R.id.endDate);
        filterButton =
findViewById(R.id.filterButton);
        barChart =
findViewById(R.id.barChart);
        sdf = new SimpleDateFormat("dd-MM-yyyy", Locale.getDefault());
        calendar = Calendar.getInstance();

        // Set today's date as default for start and end date
        String todayDate =
sdf.format(calendar.getTime());

        startDateTextView.setText(todayDate);
        endDateTextView.setText(todayDate);

        startDateTextView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                showDatePickerDialog(startDateTextView);
            }
        });

        endDateTextView.setOnClickListener(new

```

```

View.OnClickListener() {
    @Override
    public void onClick(View v) {

showDatePickerDialog(endDateTextView)
;
    }
});

filterButton.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String startDate =
startDateTextView.getText().toString();
        String endDate =
endDateTextView.getText().toString();
        loadPemasukanData(startDate,
endDate);
    }
});

// Load initial data without filter
loadPemasukanData(null, null);
}

private void
showDatePickerDialog(final TextView
dateTextView) {
    final Calendar calendar =
Calendar.getInstance();
    DatePickerDialog datePickerDialog =
new DatePickerDialog(this, new
DatePickerDialog.OnDateSetListener() {
        @Override
        public void onDateSet(DatePicker
view, int year, int monthOfYear, int
dayOfMonth) {
            calendar.set(Calendar.YEAR,
year);
            calendar.set(Calendar.MONTH,
monthOfYear);
            calendar.set(Calendar.DAY_OF_MONTH,
dayOfMonth);

```

```

dateTextView.setText(sdf.format(calendar.
getTime()));
        }
    }, calendar.get(Calendar.YEAR),
calendar.get(Calendar.MONTH),
calendar.get(Calendar.DAY_OF_MONTH))
;
    datePickerDialog.show();
}

private void loadPemasukanData(String
startDate, String endDate) {
    DatabaseReference pemasukanRef =
FirebaseDatabase.getInstance().getReferen
ce("admin/pemasukan");

pemasukanRef.addValueEventListener(ne
w ValueEventListener() {
    @Override
    public void
onDataChange(@NonNull DataSnapshot
snapshot) {
        tableLayout.removeAllViews();
        barChart.clear();

        // Add table header
        TableRow header = new
TableRow(TablePemasukanActivity.this);

header.addView(createTextView("Jenis \n
Telur"));

header.addView(createTextView("Jumlah
\n Telur"));

header.addView(createTextView("Harga \n
Satuan"));

header.addView(createTextView("Total \n
Harga"));

header.addView(createTextView("Tanggal
\n Penjualan"));
        tableLayout.addView(header);

```

```

        Map<String, Integer>
dateToTotalHargaMap = new
HashMap<>();
        ArrayList<BarEntry> barEntries
= new ArrayList<>();
        int index = 0;

        for (DataSnapshot data :
snapshot.getChildren()) {
            String jenisTelur =
data.child("jenisTelur").getValue(String.cla
ss);
            int jumlahTelur =
data.child("jumlahTelur").getValue(Integer
.class);
            int hargaSatuan =
data.child("hargaSatuan").getValue(Integer
.class);
            int totalHarga =
data.child("totalHarga").getValue(Integer.c
lass);
            String tanggalPenjualan =
data.child("tanggalPenjualan").getValue(Str
ing.class);

            try {
                if (startDate != null &&
endDate != null) {
                    Date start =
sdf.parse(startDate);
                    Date end =
sdf.parse(endDate);
                    Date tanggal =
sdf.parse(tanggalPenjualan);
                    if (tanggal.before(start) ||
tanggal.after(end)) {
                        continue;
                    }
                }
            } catch (Exception e) {
                e.printStackTrace();
            }

            TableRow row = new
TableRow(TablePemasukanActivity.this);

```

```

row.addView(createTextView(jenisTelur));

row.addView(createTextView(String.value
Of(jumlahTelur)));

row.addView(createTextView(String.value
Of(hargaSatuan)));

row.addView(createTextView(String.value
Of(totalHarga)));

row.addView(createTextView(tanggalPenj
ualan));
        tableLayout.addView(row);

        // Sum up total harga for each
date
        if
(dateToTotalHargaMap.containsKey(tangg
alPenjualan)) {

            dateToTotalHargaMap.put(tanggalPenjuala
n,
            dateToTotalHargaMap.get(tanggalPenjuala
n) + totalHarga);
        } else {

            dateToTotalHargaMap.put(tanggalPenjuala
n, totalHarga);
        }

        // Add data to BarChart
        ArrayList<String> dates = new
ArrayList<>(dateToTotalHargaMap.keySet
());
        for (int i = 0; i < dates.size();
i++) {
            barEntries.add(new
BarEntry(i,
            dateToTotalHargaMap.get(dates.get(i))));
        }

        if (!barEntries.isEmpty()) {
            BarDataSet barDataSet = new

```

```

BarDataSet(barEntries, "Total Harga");
        BarData barData = new
BarData(barDataSet);
        barChart.setData(barData);

        // Set x-axis labels
        XAxis xAxis =
barChart.getXAxis();
        xAxis.setValueFormatter(new
IndexAxisValueFormatter(dates));
        xAxis.setGranularity(1f);

xAxis.setPosition(XAxis.XAxisPosition.B
OTTOM);

        barChart.invalidate(); //
Refresh the chart
    }
}

@Override
public void
onCancelled(@NonNull DatabaseError
error) {
    // Handle database error
    }
});
}

private TextView createTextView(String
text) {
    TextView textView = new
TextView(this);
    textView.setText(text);
    textView.setPadding(8, 8, 8, 8);

textView.setBackgroundResource(R.drawa
ble.cell_border); // Apply cell border
drawable
    return textView;
}
}

```

TablePengeluaranActivity.java

```

package
com.aplikasi.apliaksiternaktelur.admin.lap
oran;

import android.app.DatePickerDialog;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;

import com.aplikasi.apliaksiternaktelur.R;
import
com.github.mikephil.charting.components.
XAxis;
import
com.github.mikephil.charting.data.BarEntr
y;
import
com.github.mikephil.charting.data.BarDat
a;
import
com.github.mikephil.charting.data.BarDat
aSet;
import
com.github.mikephil.charting.charts.BarC
hart;
import
com.github.mikephil.charting.formatter.In
dexAxisValueFormatter;
import
com.google.firebase.database.DataSnapsho
t;
import
com.google.firebase.database.DatabaseErr
or;
import
com.google.firebase.database.DatabaseRef
erence;
import

```

```

com.google.firebase.database.FirebaseData
base;
import
com.google.firebase.database.ValueEventL
istener;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.Locale;

public class TablePengeluaranActivity
extends AppCompatActivity {

    private TableLayout tableLayout;
    private TextView startDateTextView,
endDateTextView;
    private Button filterButton;
    private BarChart barChart;
    private SimpleDateFormat sdf;
    private Calendar calendar;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_table_pe
ngeluaran);

        tableLayout =
findViewById(R.id.tableLayout);
        startDateTextView =
findViewById(R.id.startDate);
        endDateTextView =
findViewById(R.id.endDate);
        filterButton =
findViewById(R.id.filterButton);
        barChart =
findViewById(R.id.barChart);
        sdf = new SimpleDateFormat("dd-
MM-yyyy", Locale.getDefault());
        calendar = Calendar.getInstance();

        // Set today's date as default for start

```

```

and end date
        String todayDate =
sdf.format(calendar.getTime());

startDateTextView.setText(todayDate);
        endDateTextView.setText(todayDate);

startDateTextView.setOnClickListener(ne
w View.OnClickListener() {
    @Override
    public void onClick(View v) {

showDatePickerDialog(startDateTextView
);
        }
    });

endDateTextView.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {

showDatePickerDialog(endDateTextView)
;
        }
    });

filterButton.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String startDate =
startDateTextView.getText().toString();
        String endDate =
endDateTextView.getText().toString();
        loadPengeluaranData(startDate,
endDate);
    }
});

// Load initial data without filter
loadPengeluaranData(null, null);
}

```

```

        private void
showDatePickerDialog(final TextView
dateTextView) {
    final Calendar calendar =
Calendar.getInstance();
    DatePickerDialog datePickerDialog =
new DatePickerDialog(this, new
DatePickerDialog.OnDateSetListener() {
        @Override
        public void onDateSet(DatePicker
view, int year, int monthOfYear, int
dayOfMonth) {
            calendar.set(Calendar.YEAR,
year);
            calendar.set(Calendar.MONTH,
monthOfYear);

calendar.set(Calendar.DAY_OF_MONTH,
dayOfMonth);

dateTextView.setText(sdf.format(calendar.
getTime()));
        }
    }, calendar.get(Calendar.YEAR),
calendar.get(Calendar.MONTH),
calendar.get(Calendar.DAY_OF_MONTH))
;
    datePickerDialog.show();
}

private void
loadPengeluaranData(String startDate,
String endDate) {
    DatabaseReference adminRef =
FirebaseDatabase.getInstance().getReferen
ce("admin");

adminRef.addValueEventListener(new
ValueEventListener() {
        @Override
        public void
onDataChange(@NonNull DataSnapshot
snapshot) {
            tableLayout.removeAllViews();
            barChart.clear();

```

```

// Add table header
TableRow header = new
TableRow(TablePengeluaranActivity.this);

header.addView(createTextView("Jenis"));

header.addView(createTextView("Nama"))
;

header.addView(createTextView("Harga"))
;

header.addView(createTextView("Jumlah"
));

header.addView(createTextView("Tanggal
"));

tableLayout.addView(header);

ArrayList<BarEntry> barEntries
= new ArrayList<>();
ArrayList<String> xLabels =
new ArrayList<>();
int index = 0;

// Process pakan data
if
(snapshot.child("pakan").exists()) {
    for (DataSnapshot pakan :
snapshot.child("pakan").getChildren()) {
        String jenisPakan =
pakan.child("jenisPakan").getValue(String.
class);

        int hargaPerKg =
pakan.child("hargaPerKg").getValue(Integ
er.class);

        int jumlahPakan =
pakan.child("jumlahPakan").getValue(Inte
ger.class);

        String tanggalPemberian =
pakan.child("tanggalPemberian").getValue
(String.class);

        try {
            if (startDate != null &&
endDate != null) {

```

```

                Date start =
sdf.parse(startDate);
                Date end =
sdf.parse(endDate);
                Date tanggal =
sdf.parse(tanggalPemberian);
                if (tanggal.before(start)
|| tanggal.after(end)) {
                    continue;
                }
            } catch (Exception e) {
                e.printStackTrace();
            }

```

```

                TableRow row = new
TableRow(TablePengeluaranActivity.this);

row.addView(createTextView("Pakan"));

row.addView(createTextView(jenisPakan))
;

row.addView(createTextView(String.valueOf(hargaPerKg)));

row.addView(createTextView(String.valueOf(jumlahPakan)));

row.addView(createTextView(tanggalPemberian));

                tableLayout.addView(row);

                int totalPakan =
hargaPerKg * jumlahPakan;
                barEntries.add(new
BarEntry(index, totalPakan));

xLabels.add(tanggalPemberian);
                index++;
            }
        }

```

```

// Process obat data
if
(snapshot.child("obat").exists()) {

```

```

                for (DataSnapshot obat :
snapshot.child("obat").getChildren()) {
                    String namaObat =
obat.child("namaObat").getValue(String.class);

                    int hargaObat =
obat.child("hargaObat").getValue(Integer.class);

                    String tanggal =
obat.child("tanggal").getValue(String.class);

                    int jumlahObat =
obat.child("jumlahObat").getValue(Integer.class);

```

```

                try {
                    if (startDate != null &&
endDate != null) {
                        Date start =
sdf.parse(startDate);
                        Date end =
sdf.parse(endDate);
                        Date tanggalObat =
sdf.parse(tanggal);
                        if
(tanggalObat.before(start) ||
tanggalObat.after(end)) {
                            continue;
                        }
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                }

```

```

                TableRow row = new
TableRow(TablePengeluaranActivity.this);

row.addView(createTextView("Obat"));

row.addView(createTextView(namaObat));

row.addView(createTextView(String.valueOf(hargaObat)));

row.addView(createTextView(String.valueOf(jumlahObat)));

```

```
row.addView(createTextView(tanggal));
    tableLayout.addView(row);
```

```
        int totalObat = hargaObat *
jumlahObat;
        barEntries.add(new
BarEntry(index, totalObat));
        xLabels.add(tanggal);
        index++;
    }
}
```

```
    // Setup BarChart
    BarDataSet barDataSet = new
BarDataSet(barEntries, "Pengeluaran");
    BarData barData = new
BarData(barDataSet);
    barChart.setData(barData);
    barChart.invalidate(); // Refresh
chart
```

```
    // Set X Axis labels
    XAxis xAxis =
barChart.getXAxis();
    xAxis.setValueFormatter(new
IndexAxisValueFormatter(xLabels));
```

```
xAxis.setPosition(XAxis.XAxisPosition.BOTTOM);
    xAxis.setGranularity(1f); //
Show each label
}
```

```
@Override
public void
onCancelled(@NonNull DatabaseError
error) {
    // Handle possible errors.
}
});
}
```

```
private TextView createTextView(String
text) {
    TextView textView = new
```

```
TextView(this);
    textView.setText(text);
    textView.setPadding(8, 8, 8, 8);
```

```
textView.setBackgroundResource(R.drawable.cell_border); // Apply cell border
drawable
    return textView;
}
}
```

OBAT

DataObat.java

```
package
com.aplikasi.apliaksiternaktelur.admin.obat;
```

```
public class DataObat {
    private String id;
    private String namaObat;
    private int hargaObat;
    private String tanggal;
    private int jumlahObat; // Tambahkan
atribut jumlahObat
```

```
    // Konstruktor default diperlukan untuk
Firebase
    public DataObat() {
        // Konstruktor default diperlukan
untuk Firebase
    }
```

```
    public DataObat(String id, String
namaObat, int hargaObat, String tanggal,
int jumlahObat) {
        this.id = id;
        this.namaObat = namaObat;
        this.hargaObat = hargaObat;
        this.tanggal = tanggal;
        this.jumlahObat = jumlahObat;
    }
```

```
    public String getId() {
        return id;
```

```

    }

    public void setId(String id) {
        this.id = id;
    }

    public String getNamaObat() {
        return namaObat;
    }

    public void setNamaObat(String
namaObat) {
        this.namaObat = namaObat;
    }

    public int getHargaObat() {
        return hargaObat;
    }

    public void setHargaObat(int
hargaObat) {
        this.hargaObat = hargaObat;
    }

    public String getTanggal() {
        return tanggal;
    }

    public void setTanggal(String tanggal) {
        this.tanggal = tanggal;
    }

    public int getJumlahObat() {
        return jumlahObat;
    }

    public void setJumlahObat(int
jumlahObat) {
        this.jumlahObat = jumlahObat;
    }
}

```

Edit_obat.java

```

package
com.aplikasi.apliksiternaktelur.admin.oba

```

```

t;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;

import com.aplikasi.apliksiternaktelur.R;
import
com.google.android.gms.tasks.OnFailureL
istener;
import
com.google.android.gms.tasks.OnSuccess
Listener;
import com.google.firebase.FirebaseApp;
import
com.google.firebase.database.DatabaseRef
erence;
import
com.google.firebase.database.FirebaseData
base;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.Locale;

public class Edit_obat extends
AppCompatActivity {

```

```

    private EditText editTextHargaObat,
editTextJumlahObat;
    private Spinner spinnerNamaObat;

```

```

private DatePicker datePicker;
private Button btnUpdate, btnHapus,
btnCancel;

private DatabaseReference
databaseReference;
private String idObat; // ID obat yang
diterima dari intent

private String[] namaObatArray,
hargaObatArray;

@SuppressWarnings("MissingInflatedId")
@Override
protected void onCreate(Bundle
savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.edit_obat);

    // Inisialisasi Firebase
    FirebaseApp.initializeApp(this);

    // Inisialisasi views
    spinnerNamaObat =
findViewById(R.id.SpinnerNamaObat);
    editTextHargaObat =
findViewById(R.id.editTextHargaObat);
    editTextJumlahObat =
findViewById(R.id.editTextJumlahObat);
    datePicker =
findViewById(R.id.datePickerTanggal);
    btnUpdate =
findViewById(R.id.btnupdate);
    btnHapus =
findViewById(R.id.btnhapus);
    btnCancel =
findViewById(R.id.btncancel);

    // Inisialisasi Firebase Database
    databaseReference =
FirebaseDatabase.getInstance().getReferen
ce().child("admin/obat");

    // Ambil array nama obat dan harga
obat dari resources
    namaObatArray =

```

```

getResources().getStringArray(R.array.na
ma_obat_array);
    hargaObatArray =
getResources().getStringArray(R.array.har
ga_obat_array);

    // Setup ArrayAdapter untuk Spinner
nama obat
    ArrayAdapter<String>
adapterNamaObat = new
ArrayAdapter<>(this,

    android.R.layout.simple_spinner_item,
namaObatArray);

    adapterNamaObat.setDropDownViewReso
urce(android.R.layout.simple_spinner_dro
pdown_item);

    spinnerNamaObat.setAdapter(adapterNam
aObat);

    // Set listener untuk Spinner nama
obat

    spinnerNamaObat.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
        @Override
        public void
onItemSelected(AdapterView<?> parent,
View view, int position, long id) {
            if (position >= 0 && position <
namaObatArray.length) {
                String hargaObat =
hargaObatArray[position];

                editTextHargaObat.setText(hargaObat); //
Set harga obat ke editTextHargaObat
            }
        }

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

```

```

        // Do nothing
    }
});

// Mendapatkan data obat yang akan
diedit dari Intent
Intent intent = getIntent();
if (intent != null) {
    idObat =
intent.getStringExtra("idObat");
    String namaObat =
intent.getStringExtra("nama");
    int hargaObat =
intent.getIntExtra("hargaObat", 0);
    int jumlahObat =
intent.getIntExtra("jumlahObat", 0);
    String tanggal =
intent.getStringExtra("tanggal");

    // Set data ke Spinner dan EditText
    if (namaObat != null) {
        int position =
getPositionFromNamaObat(namaObat);

spinnerNamaObat.setSelection(position);
    }

    editTextHargaObat.setText(String.valueOf(
hargaObat));

    editTextJumlahObat.setText(String.valueO
f(jumlahObat));

    // Mengatur tanggal pada
DatePicker
    if (tanggal != null) {
        try {
            SimpleDateFormat sdf = new
SimpleDateFormat("dd-MM-yyyy",
Locale.getDefault());
            Date date =
sdf.parse(tanggal);
            Calendar calendar =
Calendar.getInstance();
            if (date != null) {

```

```

calendar.setTime(date);

datePicker.init(calendar.get(Calendar.YEA
R), calendar.get(Calendar.MONTH),
calendar.get(Calendar.DAY_OF_MONTH),
null);
        }
    } catch (ParseException e) {
        e.printStackTrace();
    }
}

// Menangani klik tombol Update
btnUpdate.setOnClickListener(v ->
updateDataObat());

// Menangani klik tombol Hapus
btnHapus.setOnClickListener(v ->
deleteObat());

// Menangani klik tombol Batal
btnCancel.setOnClickListener(v ->
finish());
}

private int
getPositionFromNamaObat(String
namaObat) {
    for (int i = 0; i <
namaObatArray.length; i++) {
        if
(namaObatArray[i].equals(namaObat)) {
            return i;
        }
    }
    return 0; // Return default position if
not found
}

private void updateDataObat() {
    String namaObat =
spinnerNamaObat.getSelectedItem().toStri
ng();
    int hargaObat =
Integer.parseInt(editTextHargaObat.getTex

```

```

t().toString().trim());
    int jumlahObat =
Integer.parseInt(editTextJumlahObat.getText().toString().trim());

    int day =
datePicker.getDayOfMonth();
    int month = datePicker.getMonth() +
1;
    int year = datePicker.getYear();
    String tanggalPemberian = year + "-"
+ month + "-" + day;

    if (namaObat.isEmpty() ||
editTextJumlahObat.getText().toString().trim().isEmpty()) {
        Toast.makeText(this, "Harap
lengkapi semua data",
Toast.LENGTH_SHORT).show();
        return;
    }

    DataObat dataObat = new
DataObat(idObat, namaObat, hargaObat,
tanggalPemberian, jumlahObat);

    databaseReference.child(idObat).setValue(
dataObat)
        .addOnSuccessListener(aVoid -> {
            Toast.makeText(Edit_obat.this,
            "Data obat berhasil diperbarui",
            Toast.LENGTH_SHORT).show();
            finish();
        })
        .addOnFailureListener(e -> {
            Toast.makeText(Edit_obat.this,
            "Gagal memperbarui data: " +
            e.getMessage(),
            Toast.LENGTH_SHORT).show();
        });
    }

    private void deleteObat() {
        if (idObat != null) {

```

```

databaseReference.child(idObat).removeValue()
        .addOnSuccessListener(aVoid -> {
            Toast.makeText(Edit_obat.this, "Data obat
            berhasil dihapus",
            Toast.LENGTH_SHORT).show();
            finish();
        })
        .addOnFailureListener(e -> {
            Toast.makeText(Edit_obat.this, "Gagal
            menghapus data: " + e.getMessage(),
            Toast.LENGTH_SHORT).show();
        });
    }
}

```

ObatActivity.java

```

package
com.aplikasi.apliaksiternaktelur.admin.obat;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;
import
androidx.recyclerview.widget.LinearLayoutManager;
import

```

```
androidx.recyclerview.widget.RecyclerView;
```

```
import
com.aplikasi.apliaksiternaktelur.MainActivity;
import com.aplikasi.apliaksiternaktelur.R;
import
com.google.firebase.database.DataSnapshot;
import
com.google.firebase.database.DatabaseError;
import
com.google.firebase.database.DatabaseReference;
import
com.google.firebase.database.FirebaseDatabase;
import
com.google.firebase.database.ValueEventListener;
```

```
import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;
```

```
public class ObatActivity extends
    AppCompatActivity {
```

```
    private RecyclerView
recyclerViewObat;
    private ObatAdapter obatAdapter;
    private List<DataObat> dataObatList;
    private List<DataObat> filteredList;
    private EditText txtCari;
    private TextView txtTotalPengeluaran;
    private DatabaseReference adminRef;
    private ImageButton btnEdit;
```

```
    @SuppressWarnings("MissingInflatedId")
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.data_obat);
```

```
        // Inisialisasi views
        recyclerViewObat =
findViewById(R.id.recyclerViewObat);
        txtCari = findViewById(R.id.txtCari);
        txtTotalPengeluaran =
findViewById(R.id.txtJumlahPengeluaran);
        btnEdit =
findViewById(R.id.btnEdit); // Pastikan ID
ini sesuai dengan XML
```

```
        // RecyclerView setup
```

```
        recyclerViewObat.setLayoutManager(new
LinearLayoutManager(this));
        dataObatList = new ArrayList<>();
        filteredList = new ArrayList<>();
        obatAdapter = new
ObatAdapter(filteredList);
```

```
        recyclerViewObat.setAdapter(obatAdapter);
```

```
        // Firebase setup
```

```
        FirebaseDatabase database =
FirebaseDatabase.getInstance();
        adminRef =
database.getReference("admin/obat");
```

```
        // Fetch data from Firebase
        fetchDataObat();
```

```
        // Search functionality
```

```
        txtCari.addTextChangedListener(new
TextWatcher() {
            @Override
            public void
beforeTextChanged(CharSequence s, int
start, int count, int after) {
                // Do nothing
            }
        })
```

```
        @Override
        public void
```

```

onTextChanged(CharSequence s, int start,
int before, int count) {
    filterData(s.toString());
}

@Override
public void
afterTextChanged(Editable s) {
    // Do nothing
}
});

// Back button click listener
ImageView ivBack =
findViewById(R.id.ivBack);
ivBack.setOnClickListener(v ->
onBackPressed());

// Edit button click listener
if (btnEdit != null) {
    btnEdit.setOnClickListener(v -> {
        Intent intent = new
Intent(ObatActivity.this, Edit_obat.class);
        startActivity(intent);
    });
} else {
    Log.e("ObatActivity", "btnEdit is
null. Check the XML layout for the correct
ID.");
}
}

// Method to start tambah_obat activity
public void tambah_obat(View view) {
    Intent intent = new Intent(this,
tambah_obat.class);
    startActivity(intent);
}

// Fetch data from Firebase
private void fetchDataObat() {

adminRef.addValueEventListener(new
ValueEventListener() {
    @Override
    public void

```

```

onDataChange(@NonNull DataSnapshot
dataSnapshot) {
    dataObatList.clear();
    filteredList.clear(); // Clear the
filtered list

    for (DataSnapshot snapshot :
dataSnapshot.getChildren()) {
        String id = snapshot.getKey();
        String namaObat =
snapshot.child("namaObat").getValue(Stri
ng.class);
        String tanggal =
snapshot.child("tanggal").getValue(String.
class);
        Integer hargaObatInteger =
snapshot.child("hargaObat").getValue(Inte
ger.class);
        Integer jumlahObatInteger =
snapshot.child("jumlahObat").getValue(Int
eger.class);

        int hargaObat =
(hargaObatInteger != null) ?
hargaObatInteger : 0;
        int jumlahObat =
(jumlahObatInteger != null) ?
jumlahObatInteger : 0;

        DataObat dataObat = new
DataObat(id, namaObat, hargaObat,
tanggal, jumlahObat);
        dataObatList.add(dataObat);
    }

    filteredList.addAll(dataObatList);

    obatAdapter.notifyDataSetChanged();
    calculateTotalPengeluaran();
}

@Override
public void
onCancelled(@NonNull DatabaseError
databaseError) {

```

```

        Log.e("Firebase", "Error reading
data from Firebase",
databaseError.toException());
    }
});
}

```

```

// Calculate total pengeluaran based on
dataObatList
private void calculateTotalPengeluaran()
{

```

```

    int totalPengeluaran = 0;

    for (DataObat data : dataObatList) {
        totalPengeluaran +=
data.getHargaObat() *
data.getJumlahObat();
    }

```

```

    Locale localeID = new Locale("id",
"ID");
    NumberFormat formatRupiah =
NumberFormat.getCurrencyInstance(local
eID);
    String formattedTotalPengeluaran =
formatRupiah.format(totalPengeluaran);

```

```

txtTotalPengeluaran.setText(formattedTota
lPengeluaran);
}

```

```

// Filter data based on search query
private void filterData(String query) {
    filteredList.clear();
    for (DataObat data : dataObatList) {
        if
(data.getNamaObat().toLowerCase().conta
ins(query.toLowerCase())) {
            filteredList.add(data);
        }
    }
    obatAdapter.notifyDataSetChanged();
}

```

```

// Back button press

```

```

public void pressback(View view) {
    Intent intent = new Intent(this,
MainActivity.class);
    startActivity(intent);
    finish();
}

```

```

public void btn_detailobat(View view) {
    Intent intent = new Intent(this,
Table_obat.class);
    startActivity(intent);
}
}

```

ObatAdapter.java

```

package
com.aplikasi.apliaksiternaktelur.admin.oba
t;

```

```

import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageButton;
import android.widget.TextView;

```

```

import androidx.annotation.NonNull;
import
androidx.recyclerview.widget.RecyclerVie
w;

```

```

import com.aplikasi.apliaksiternaktelur.R;

```

```

import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

```

```

public class ObatAdapter extends
RecyclerView.Adapter<ObatAdapter.Obat
ViewHolder> {

```

```

    private List<DataObat> dataObatList;
    private List<DataObat>
originalDataObatList; // To store original

```

data for filtering

```
public ObatAdapter(List<DataObat>
dataObatList) {
    this.dataObatList = dataObatList;
    this.originalDataObatList = new
ArrayList<>(dataObatList); // Make a
copy of original data
}
```

```
@NonNull
@Override
public ObatViewHolder
onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
    View view =
LayoutInflater.from(parent.getContext()).in
flate(R.layout.item_obat, parent, false);
    return new ObatViewHolder(view);
}
```

```
@Override
public void
onBindViewHolder(@NonNull
ObatViewHolder holder, int position) {
    DataObat dataObat =
dataObatList.get(position);
```

```
    holder.txtNamaObat.setText("" +
dataObat.getNamaObat());
    holder.jumlah.setText("" +
dataObat.getJumlahObat());
    holder.tanggal.setText("" +
dataObat.getTanggal());
```

```
    holder.txtHargaObat.setText(formatCurren
cy(dataObat.getHargaObat()));
}
```

```
private String formatCurrency(int
amount) {
    Locale localeID = new Locale("in",
"ID");
    NumberFormat formatRupiah =
NumberFormat.getCurrencyInstance(local
eID);
```

```
    return formatRupiah.format(amount);
}
```

```
@Override
public int getItemCount() {
    return dataObatList.size();
}
```

```
public void filterList(List<DataObat>
filteredList) {
    dataObatList = filteredList;
    notifyDataSetChanged();
}
```

```
public class ObatViewHolder extends
RecyclerView.ViewHolder {
    TextView txtUmurAyam,
txtNamaObat, txtHargaObat, tanggal,
jumlah;
    ImageButton btnEdit;
```

```
    public ObatViewHolder(@NonNull
View itemView) {
        super(itemView);
        txtNamaObat =
itemView.findViewById(R.id.txtNamaOba
t);
        txtHargaObat =
itemView.findViewById(R.id.txtHargaOb
at);
        tanggal =
itemView.findViewById(R.id.txtTanggalP
emberian);
        jumlah =
itemView.findViewById(R.id.txtJumlahOb
at);
        btnEdit =
itemView.findViewById(R.id.btnEdit);
```

```
        btnEdit.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                int position =
getAdapterPosition();
                if (position !=
```

```

RecyclerView.NO_POSITION) {
    DataObat dataObat =
ObatAdapter.this.dataObatList.get(position
);
    Intent intent = new
Intent(v.getContext(), Edit_obat.class);
    intent.putExtra("idObat",
dataObat.getId());
    intent.putExtra("nama",
dataObat.getNamaObat());

intent.putExtra("hargaObat",
dataObat.getHargaObat());

intent.putExtra("jumlahObat",
dataObat.getJumlahObat());
    intent.putExtra("tanggal",
dataObat.getTanggal()); // Make sure date
is in yyyy-MM-dd format

v.getContext().startActivity(intent);
    }
    }
    });
    }
}
}

```

Table_obat.java

```

package
com.aplikasi.apliksiternaktelur.admin.oba
t;

import android.os.Bundle;
import android.view.View;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;

```

```

import com.aplikasi.apliksiternaktelur.R;
import
com.google.firebase.database.DataSnapsho
t;
import
com.google.firebase.database.DatabaseErr
or;
import
com.google.firebase.database.DatabaseRef
erence;
import
com.google.firebase.database.FirebaseData
base;
import
com.google.firebase.database.ValueEventL
istener;

```

```

public class Table_obat extends
AppCompatActivity {

```

```

    private TableLayout tableLayoutObat;
    private DatabaseReference
databaseReference;

```

```

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

setContentView(R.layout.activity_table_ob
at);

```

```

    // Inisialisasi Firebase Database
    databaseReference =
FirebaseDatabase.getInstance().getReferen
ce().child("admin/obat");

```

```

    // Inisialisasi views
    tableLayoutObat =
findViewById(R.id.tableLayoutObat);

```

```

    // Load data obat
    loadData();
}

```

```

private void loadData() {

```

```

databaseReference.addValueEventListener
(new ValueEventListener() {
    @Override
    public void
onDataChange(@NonNull DataSnapshot
dataSnapshot) {
    // Clear existing rows in the table

tableLayoutObat.removeAllViews();

    // Create table headers
    TableRow headerRow = new
TableRow(Table_obat.this);
    String[] headers = {"Tanggal",
"Nama Obat", "Jumlah", "Harga",
"Total"};
    for (String header : headers) {
        TextView textView = new
TextView(Table_obat.this);
        textView.setText(header);
        textView.setPadding(8, 8, 8,
8);

headerRow.addView(textView);
    }

tableLayoutObat.addView(headerRow);

    // Populate table with data
    for (DataSnapshot snapshot :
dataSnapshot.getChildren()) {
        DataObat dataObat =
snapshot.getValue(DataObat.class);
        if (dataObat != null) {
            TableRow row = new
TableRow(Table_obat.this);
            addTextViewToRow(row,
dataObat.getTanggal());
            addTextViewToRow(row,
dataObat.getNamaObat());
            addTextViewToRow(row,
String.valueOf(dataObat.getJumlahObat())
);
            addTextViewToRow(row,
String.valueOf(dataObat.getHargaObat()));

```

```

            addTextViewToRow(row,
String.valueOf(dataObat.getJumlahObat()
* dataObat.getHargaObat()));

tableLayoutObat.addView(row);
        }
    }

    @Override
    public void
onCancelled(@NonNull DatabaseError
databaseError) {
        Toast.makeText(Table_obat.this,
"Gagal mengambil data: " +
databaseError.getMessage(),
Toast.LENGTH_SHORT).show();
    }
});
}

private void
addTextViewToRow(TableRow row,
String text) {
    TextView textView = new
TextView(Table_obat.this);
    textView.setText(text);
    textView.setPadding(8, 8, 8, 8);
    row.addView(textView);
}
}

```

tambah_obat.java

```

package
com.aplikasi.apliaksiternaktelur.admin.oba
t;

import android.os.Bundle;
import android.view.View;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;

```

```
import
androidx.appcompat.app.AppCompatActivity;
```

```
import com.aplikasi.apliaksiternaktelur.R;
import
com.google.firebase.database.DataSnapsho
t;
import
com.google.firebase.database.DatabaseErr
or;
import
com.google.firebase.database.DatabaseRef
erence;
import
com.google.firebase.database.FirebaseData
base;
import
com.google.firebase.database.ValueEventL
istener;
```

```
public class Table_obat extends
AppCompatActivity {
```

```
    private TableLayout tableLayoutObat;
    private DatabaseReference
databaseReference;
```

```
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_table_ob
at);
```

```
        // Inisialisasi Firebase Database
        databaseReference =
FirebaseDatabase.getInstance().getReferen
ce().child("admin/obat");
```

```
        // Inisialisasi views
        tableLayoutObat =
findViewById(R.id.tableLayoutObat);
```

```
        // Load data obat
```

```
        loadData();
    }
```

```
    private void loadData() {
```

```
        databaseReference.addValueEventListener
(new ValueEventListener() {
```

```
            @Override
```

```
            public void
```

```
onDataChange(@NonNull DataSnapshot
dataSnapshot) {
```

```
                // Clear existing rows in the table
```

```
                tableLayoutObat.removeAllViews();
```

```
                // Create table headers
```

```
                TableRow headerRow = new
```

```
                TableRow(Table_obat.this);
```

```
                String[] headers = {"Tanggal",
"Nama Obat", "Jumlah", "Harga",
"Total"};
```

```
                for (String header : headers) {
```

```
                    TextView textView = new
```

```
                    TextView(Table_obat.this);
```

```
                    textView.setText(header);
```

```
                    textView.setPadding(8, 8, 8,
```

```
                    8);
```

```
                    headerRow.addView(textView);
```

```
                }
```

```
                tableLayoutObat.addView(headerRow);
```

```
                // Populate table with data
```

```
                for (DataSnapshot snapshot :
```

```
dataSnapshot.getChildren()) {
```

```
                    DataObat dataObat =
```

```
                    snapshot.getValue(DataObat.class);
```

```
                    if (dataObat != null) {
```

```
                        TableRow row = new
```

```
                        TableRow(Table_obat.this);
```

```
                        addTextViewToRow(row,
```

```
                        dataObat.getTanggal());
```

```
                        addTextViewToRow(row,
```

```
                        dataObat.getNamaObat());
```

```
                        addTextViewToRow(row,
```

```

String.valueOf(dataObat.getJumlahObat())
);
        addTextViewToRow(row,
String.valueOf(dataObat.getHargaObat()));
        addTextViewToRow(row,
String.valueOf(dataObat.getJumlahObat()
* dataObat.getHargaObat()));

tableLayoutObat.addView(row);
    }
}

@Override
public void
onCancelled(@NonNull DatabaseError
databaseError) {
    Toast.makeText(Table_obat.this,
"Gagal mengambil data: " +
databaseError.getMessage(),
Toast.LENGTH_SHORT).show();
}
});
}

private void
addTextViewToRow(TableRow row,
String text) {
    TextView textView = new
TextView(Table_obat.this);
    textView.setText(text);
    textView.setPadding(8, 8, 8, 8);
    row.addView(textView);
}
}

```

OPERASIONAL

DataOperasional.java

```

package
com.aplikasi.apliaksiternaktelur.admin.ope
rasional;

public class DataOperasional {
    private String id;

```

```

private String tanggal;
private String namaAlat;
private int jumlahUnit;
private double harga;
private int lamaPemakaian;

// Konstruktor tanpa argumen diperlukan
oleh Firebase
public DataOperasional() {
    // Diperlukan oleh Firebase
}

// Konstruktor dengan parameter
public DataOperasional(String id, String
tanggal, String namaAlat, int jumlahUnit,
double harga, int lamaPemakaian) {
    this.id = id;
    this.tanggal = tanggal;
    this.namaAlat = namaAlat;
    this.jumlahUnit = jumlahUnit;
    this.harga = harga;
    this.lamaPemakaian =
lamaPemakaian;
}

// Getter dan Setter
public String getId() {
    return id;
}

public void setId(String id) {
    this.id = id;
}

public String getTanggal() {
    return tanggal;
}

public void setTanggal(String tanggal) {
    this.tanggal = tanggal;
}

public String getNamaAlat() {
    return namaAlat;
}

```

```

    public void setNamaAlat(String
namaAlat) {
        this.namaAlat = namaAlat;
    }

    public int getJumlahUnit() {
        return jumlahUnit;
    }

    public void setJumlahUnit(int
jumlahUnit) {
        this.jumlahUnit = jumlahUnit;
    }

    public double getHarga() {
        return harga;
    }

    public void setHarga(double harga) {
        this.harga = harga;
    }

    public int getLamaPemakaian() {
        return lamaPemakaian;
    }

    public void setLamaPemakaian(int
lamaPemakaian) {
        this.lamaPemakaian =
lamaPemakaian;
    }
}

```

EditOperasionalActivity.java

```

package
com.aplikasi.apliksiternaktelur.admin.ope
rasional;

import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

```

```

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;

import com.aplikasi.apliksiternaktelur.R;
import
com.google.firebase.database.DataSnapsho
t;
import
com.google.firebase.database.DatabaseErr
or;
import
com.google.firebase.database.DatabaseRef
erence;
import
com.google.firebase.database.FirebaseData
base;
import
com.google.firebase.database.ValueEventL
istener;

```

```

public class EditOperasionalActivity
extends AppCompatActivity {

```

```

    private EditText edtTanggal,
    edtNamaAlat, edtJumlahUnit, edtHarga,
    edtLamaPemakaian;
    private Button btnUpdate;
    private DatabaseReference
operasionalRef;
    private String operasionalId;

```

```

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_edit_ope
rasional);

```

```

        // Inisialisasi views
        edtTanggal =
findViewById(R.id.edtTanggal);
        edtNamaAlat =
findViewById(R.id.edtNamaAlat);

```

```

        edtJumlahUnit =
findViewById(R.id.edtJumlahUnit);
        edtHarga =
findViewById(R.id.edtHarga);
        edtLamaPemakaian =
findViewById(R.id.edtLamaPemakaian);
        btnUpdate =
findViewById(R.id.btnUpdate);

```

```

        // Firebase setup
        FirebaseDatabase database =
FirebaseDatabase.getInstance();
        operasionalRef =
database.getReference("admin/operasional
");

```

```

        // Get operasional ID from intent
        operasionalId =
getIntent().getStringExtra("id");

```

```

        // Load data for editing
        loadOperasionalData();

```

```

        // Update button click listener
        btnUpdate.setOnClickListener(v ->
updateDataOperasional());
    }

```

```

    private void loadOperasionalData() {

operasionalRef.child(operasionalId).addVa
lueEventListener(new
ValueEventListener() {
    @Override
    public void
onDataChange(@NonNull DataSnapshot
dataSnapshot) {
        DataOperasional data =
dataSnapshot.getValue(DataOperasional.cl
ass);
        if (data != null) {

edtTanggal.setText(data.getTanggal());

```

```

edtNamaAlat.setText(data.getNamaAlat());

```

```

edtJumlahUnit.setText(String.valueOf(data
.getJumlahUnit()));

```

```

edtHarga.setText(String.valueOf(data.getH
arga()));

```

```

edtLamaPemakaian.setText(String.valueOf
(data.getLamaPemakaian()));
    }
}

```

```

    @Override
    public void
onCancelled(@NonNull DatabaseError
databaseError) {
        // Handle possible errors.
    }
});
}

```

```

    private void updateDataOperasional() {
        String tanggal =
edtTanggal.getText().toString().trim();
        String namaAlat =
edtNamaAlat.getText().toString().trim();
        String jumlahUnitStr =
edtJumlahUnit.getText().toString().trim();
        String hargaStr =
edtHarga.getText().toString().trim();
        String lamaPemakaianStr =
edtLamaPemakaian.getText().toString().tri
m();

```

```

        if (TextUtils.isEmpty(tanggal) ||
TextUtils.isEmpty(namaAlat) ||
TextUtils.isEmpty(jumlahUnitStr) ||
TextUtils.isEmpty(hargaStr) ||
TextUtils.isEmpty(lamaPemakaianStr)) {
            Toast.makeText(this, "Semua field
harus diisi",
Toast.LENGTH_SHORT).show();
            return;
        }

```

```

        int jumlahUnit =
Integer.parseInt(jumlahUnitStr);

```

```

        double harga =
Integer.parseInt(hargaStr);
        int lamaPemakaian =
Integer.parseInt(lamaPemakaianStr);

        DataOperasional dataOperasional =
new DataOperasional(operasionalId,
tanggal, namaAlat, jumlahUnit, harga,
lamaPemakaian);

operasionalRef.child(operasionalId).setValue(dataOperasional)
        .addOnSuccessListener(aVoid ->
{

    Toast.makeText(EditOperasionalActivity.this, "Data berhasil diperbarui",
    Toast.LENGTH_SHORT).show();
        finish();
    })
        .addOnFailureListener(e -> {

    Toast.makeText(EditOperasionalActivity.this, "Gagal memperbarui data",
    Toast.LENGTH_SHORT).show();
        });
    }
}

```

OperasionalActivity.java

```

package
com.aplikasi.apliaksiternaktelur.admin.operasional;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;

```

```

import
androidx.recyclerview.widget.LinearLayoutManager;
import
androidx.recyclerview.widget.RecyclerView;

import
com.aplikasi.apliaksiternaktelur.MainActivity;
import com.aplikasi.apliaksiternaktelur.R;
import
com.google.firebase.database.DataSnapshot;
import
com.google.firebase.database.DatabaseError;
import
com.google.firebase.database.DatabaseReference;
import
com.google.firebase.database.FirebaseDatabase;
import
com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;
import java.util.List;
public class OperasionalActivity extends
AppCompatActivity {

```

```

    private RecyclerView
recyclerViewOperasional;
    private OperasionalAdapter
operasionalAdapter;
    private List<DataOperasional>
dataOperasionalList;
    private DatabaseReference
operasionalRef;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```
setContentView(R.layout.activity_operasio
nal);
```

```
recyclerViewOperasional =
findViewById(R.id.recyclerViewOperasio
nal);
```

```
ImageView ivTambah =
findViewById(R.id.ivTambah);
```

```
recyclerViewOperasional.setLayoutManager(
new LinearLayoutManager(this));
dataOperasionalList = new
ArrayList<>();
operasionalAdapter = new
OperasionalAdapter(dataOperasionalList,
this);
```

```
recyclerViewOperasional.setAdapter(opera
sionalAdapter);
```

```
FirebaseDatabase database =
FirebaseDatabase.getInstance();
operasionalRef =
database.getReference("admin/operasional
");
```

```
fetchDataOperasional();
```

```
ivTambah.setOnClickListener(v -> {
Intent intent = new
Intent(OperasionalActivity.this,
TambahOperasionalActivity.class);
startActivity(intent);
});
}
```

```
private void fetchDataOperasional() {
```

```
operasionalRef.addValueEventListener(ne
w ValueEventListener() {
@Override
public void
onDataChange(@NonNull DataSnapshot
dataSnapshot) {
dataOperasionalList.clear();
```

```
for (DataSnapshot snapshot :
dataSnapshot.getChildren()) {
DataOperasional
dataOperasional =
snapshot.getValue(DataOperasional.class);

dataOperasionalList.add(dataOperasional);
}
```

```
operasionalAdapter.notifyDataSetChanged()
};
```

```
@Override
public void
onCancelled(@NonNull DatabaseError
databaseError) {
```

```
Toast.makeText(OperasionalActivity.this,
"Gagal memuat data",
Toast.LENGTH_SHORT).show();
}
});
}
```

```
public void pressback(View view) {
Intent intent = new Intent(this,
MainActivity.class);
```

```
intent.addFlags(Intent.FLAG_ACTIVITY_
CLEAR_TOP |
Intent.FLAG_ACTIVITY_NEW_TASK);
startActivity(intent);
finish();
}
}
```

OperasionalAdapter.java

```
package
com.aplikasi.apliaksiternaktelur.admin.ope
rasional;
```

```
import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
```

```

import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import
androidx.recyclerview.widget.RecyclerView;

import com.aplikasi.apliaksiternaktelur.R;

import java.util.List;

public class OperasionalAdapter extends
RecyclerView.Adapter<RecyclerView.View
ViewHolder> {

    private static final int
VIEW_TYPE_HEADER = 0;
    private static final int
VIEW_TYPE_ITEM = 1;

    private List<DataOperasional>
dataOperasionalList;
    private Context context;

    public
OperasionalAdapter(List<DataOperasional
> dataOperasionalList, Context context) {
        this.dataOperasionalList =
dataOperasionalList;
        this.context = context;
    }

    @NonNull
    @Override
    public RecyclerView.ViewHolder
onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
        if (viewType ==
VIEW_TYPE_HEADER) {
            View view =
LayoutInflater.from(parent.getContext())

.inflate(R.layout.header_operasional,

```

```

parent, false);
            return new
HeaderViewHolder(view);
        } else {
            View view =
LayoutInflater.from(parent.getContext())

.inflate(R.layout.item_operasional, parent,
false);
            return new
OperasionalViewHolder(view);
        }
    }

    @Override
    public void
onBindViewHolder(@NonNull
RecyclerView.ViewHolder holder, int
position) {
        if (holder instanceof
OperasionalViewHolder) {
            DataOperasional data =
dataOperasionalList.get(position - 1); //
Adjust for header
            OperasionalViewHolder
operasionalViewHolder =
(OperasionalViewHolder) holder;

operasionalViewHolder.txtTanggal.setText
(data.getTanggal());

operasionalViewHolder.txtNamaAlat.setTe
xt(data.getNamaAlat());

operasionalViewHolder.txtJumlahUnit.setT
ext(String.valueOf(data.getJumlahUnit()));

operasionalViewHolder.txtHarga.setText(S
tring.valueOf(data.getHarga()));

operasionalViewHolder.txtLamaPemakaia
n.setText(String.valueOf(data.getLamaPe
makaian()));

operasionalViewHolder.ivEdit.setOnClick

```

```

Listener(v -> {
    Intent intent = new
Intent(context,
EditOperasionalActivity.class);
    intent.putExtra("id",
data.getId());
    context.startActivity(intent);
});

operasionalViewHolder.ivDelete.setOnClickListener(v -> {
    // Implement delete functionality
});
}

@Override
public int getItemCount() {
    return dataOperasionalList.size() + 1;
// +1 for header
}

@Override
public int getItemViewType(int
position) {
    return (position == 0) ?
VIEW_TYPE_HEADER :
VIEW_TYPE_ITEM;
}

public static class HeaderViewHolder
extends RecyclerView.ViewHolder {
    public HeaderViewHolder(@NonNull
View itemView) {
        super(itemView);
    }
}

public static class
OperasionalViewHolder extends
RecyclerView.ViewHolder {
    TextView txtTanggal, txtNamaAlat,
txtJumlahUnit, txtHarga,
txtLamaPemakaian;
    ImageView ivEdit, ivDelete;

```

```

public
OperasionalViewHolder(@NonNull View
itemView) {
    super(itemView);
    txtTanggal =
itemView.findViewById(R.id.txtTanggal);
    txtNamaAlat =
itemView.findViewById(R.id.txtNamaAlat
);
    txtJumlahUnit =
itemView.findViewById(R.id.txtJumlahUn
it);
    txtHarga =
itemView.findViewById(R.id.txtHarga);
    txtLamaPemakaian =
itemView.findViewById(R.id.txtLamaPem
akaian);
    ivEdit =
itemView.findViewById(R.id.ivEdit);
    ivDelete =
itemView.findViewById(R.id.ivDelete);
}
}

```

TambahOperasionalActivity.java

```

package
com.aplikasi.apliksiternaktelur.admin.ope
rasional;

import android.app.DatePickerDialog;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import
androidx.appcompat.app.AppCompatActivity;

```

```
import com.aplikasi.apliaksiternaktelur.R;
import
com.google.firebase.database.DatabaseRef
erence;
import
com.google.firebase.database.FirebaseData
base;
```

```
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;
import java.util.UUID;
```

```
public class TambahOperasionalActivity
extends AppCompatActivity {
```

```
    private EditText edtNamaAlat,
    edtJumlahUnit, edtHarga,
    edtLamaPemakaian;
```

```
    private TextView tvDatePicker;
    private Button btnSimpan;
    private DatabaseReference
    operasionalRef;
```

```
    @Override
    protected void onCreate(Bundle
    savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_tambah_
    operasional);
```

```
        // Inisialisasi views
        tvDatePicker =
        findViewById(R.id.tvDatePicker);
        edtJumlahUnit =
        findViewById(R.id.edtJumlahUnit);
        edtNamaAlat =
        findViewById(R.id.edtNamaAlat);
        edtHarga =
        findViewById(R.id.edtHarga);
        edtLamaPemakaian =
        findViewById(R.id.edtLamaPemakaian);
        btnSimpan =
        findViewById(R.id.btnSimpan);
```

```
        // Firebase setup
        FirebaseDatabase database =
        FirebaseDatabase.getInstance();
        operasionalRef =
        database.getReference("admin/operasional
        ");
```

```
        // Simpan data button click listener
        btnSimpan.setOnClickListener(v ->
        simpanDataOperasional());
```

```
        SimpleDateFormat sdf = new
        SimpleDateFormat("dd-MM-yyyy",
        Locale.getDefault());
        String currentDate =
        sdf.format(Calendar.getInstance().getTime
        ());
```

```
        tvDatePicker.setText(currentDate);
```

```
        tvDatePicker.setOnClickListener(new
        View.OnClickListener() {
```

```
            @Override
            public void onClick(View v) {
```

```
                Calendar calendar =
                Calendar.getInstance();
                int year =
                calendar.get(Calendar.YEAR);
                int month =
                calendar.get(Calendar.MONTH);
                int day =
                calendar.get(Calendar.DAY_OF_MONTH);
```

```
                DatePickerDialog
                datePickerDialog = new
                DatePickerDialog(TambahOperasionalActi
                vity.this,
```

```
                    new
                    DatePickerDialog.OnDateSetListener() {
                        @Override
                        public void
                        onDateSet(DatePicker view, int year, int
                        month, int dayOfMonth) {
                            Calendar selectedDate
                            = Calendar.getInstance();
                            selectedDate.set(year,
                            month, dayOfMonth);
```

```

        String date =
sdf.format(selectedDate.getTime());

tvDatePicker.setText(date);
    }
    }, year, month, day);
    datePickerDialog.show();
    }
});
}

private void simpanDataOperasional() {
    String tanggal =
tvDatePicker.getText().toString().trim();
    String namaAlat =
edtNamaAlat.getText().toString().trim();
    String jumlahUnitStr =
edtJumlahUnit.getText().toString().trim();
    String hargaStr =
edtHarga.getText().toString().trim();
    String lamaPemakaianStr =
edtLamaPemakaian.getText().toString().trim();

    if (TextUtils.isEmpty(tanggal) ||
TextUtils.isEmpty(namaAlat) ||
TextUtils.isEmpty(jumlahUnitStr) ||
TextUtils.isEmpty(hargaStr) ||
TextUtils.isEmpty(lamaPemakaianStr)) {
        Toast.makeText(this, "Semua field
harus diisi",
Toast.LENGTH_SHORT).show();
        return;
    }

    try {
        int jumlahUnit =
Integer.parseInt(jumlahUnitStr);
        double harga =
Double.parseDouble(hargaStr);
        int lamaPemakaian =
Integer.parseInt(lamaPemakaianStr);

        String id =
UUID.randomUUID().toString();
        DataOperasional dataOperasional =

```

```

new DataOperasional(id, tanggal,
namaAlat, jumlahUnit, harga,
lamaPemakaian);

operasionalRef.child(id).setValue(dataOperasional)
        .addOnSuccessListener(aVoid
-> {

    Toast.makeText(TambahOperasionalActivit
y.this, "Data berhasil disimpan",
Toast.LENGTH_SHORT).show();
        finish();
    })
        .addOnFailureListener(e -> {

    Toast.makeText(TambahOperasionalActivit
y.this, "Gagal menyimpan data",
Toast.LENGTH_SHORT).show();
        });
    } catch (NumberFormatException e)
    {
        Toast.makeText(this, "Format harga
atau jumlah unit tidak valid",
Toast.LENGTH_SHORT).show();
    }
}
}

```

PAKAN

DataPakan.java

```

package
com.aplikasi.apliaksiternaktelur.admin.pak
an;

```

```

public class DataPakan {
    private String id;
    private String nama;
    private int hargaPerKg;
    private int jumlahPakan;
    private String keteranganPemberian;
    private String tanggalPemberian;
    private String jenisPakan; // Tambahkan

```

jenisPakan ke dalam kelas DataPakan

```
public DataPakan() {  
    // Diperlukan konstruktor kosong  
    untuk Firebase  
}  
  
public DataPakan(String id, String  
nama, int hargaPerKg, int jumlahPakan,  
String keteranganPemberian, String  
tanggalPemberian, String jenisPakan) {  
    this.id = id;  
    this.nama = nama;  
    this.hargaPerKg = hargaPerKg;  
    this.jumlahPakan = jumlahPakan;  
    this.keteranganPemberian =  
keteranganPemberian;  
    this.tanggalPemberian =  
tanggalPemberian;  
    this.jenisPakan = jenisPakan;  
}  
  
public String getId() {  
    return id;  
}  
  
public void setId(String id) {  
    this.id = id;  
}  
  
public String getNama() {  
    return nama;  
}  
  
public void setNama(String nama) {  
    this.nama = nama;  
}  
  
public int getHargaPerKg() {  
    return hargaPerKg;  
}  
  
public void setHargaPerKg(int  
hargaPerKg) {  
    this.hargaPerKg = hargaPerKg;  
}
```

```
public int getJumlahPakan() {  
    return jumlahPakan;  
}  
  
public void setJumlahPakan(int  
jumlahPakan) {  
    this.jumlahPakan = jumlahPakan;  
}  
  
public String getKeteranganPemberian()  
{  
    return keteranganPemberian;  
}  
  
public void  
setKeteranganPemberian(String  
keteranganPemberian) {  
    this.keteranganPemberian =  
keteranganPemberian;  
}  
  
public String getTanggalPemberian() {  
    return tanggalPemberian;  
}  
  
public void setTanggalPemberian(String  
tanggalPemberian) {  
    this.tanggalPemberian =  
tanggalPemberian;  
}  
  
public String getJenisPakan() {  
    return jenisPakan;  
}  
  
public void setJenisPakan(String  
jenisPakan) {  
    this.jenisPakan = jenisPakan;  
}  
}
```

Dialog_edit.java

```
package  
com.aplikasi.apliaksiternaktelur.admin.pak
```

```

an;

import android.annotation.SuppressLint;
import android.os.Bundle;

import androidx.fragment.app.Fragment;
import
androidx.recyclerview.widget.RecyclerView;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import com.aplikasi.apliaksiternaktelur.R;

import java.net.CookieHandler;
import java.util.ArrayList;
import java.util.List;

// Di dalam aktivitas atau fragmen
public class Dialog_edit extends Fragment
{

    private RecyclerView recyclerView;
    private PakanAdapter adapter;
    private List<DataPakan> dataPakanList;
    @Override
    public View
onCreateView(LayoutInflater inflater,
ViewGroup container,
Bundle
savedInstanceState) {
    View view =
inflater.inflate(R.layout.dialog_edit_delete
_pakan, container, false);

    ArrayList<DataPakan> dataPakanList
= new ArrayList<>();

    adapter = new
PakanAdapter(dataPakanList);
    recyclerView.setAdapter(adapter);

```

```

// Implementasi listener untuk
menampilkan dialog edit
    adapter.setOnEditClickListener(new
PakanAdapter.OnEditClickListener() {
        @Override
        public void onEditClick(int
position) {
            // Panggil method untuk
menampilkan dialog edit
            showEditDialog(position);
        }
    });

    return view;
}

private void showEditDialog(int
position) {
    // Implementasi untuk menampilkan
dialog edit disini
    // Anda bisa menggunakan
AlertDialog atau DialogFragment untuk
implementasi dialog edit
    // Contoh:
    DataPakan dataPakan =
dataPakanList.get(position);
    // Lakukan sesuatu dengan dataPakan,
misalnya tampilkan dialog edit
}
}

```

Edit_pakan.java

```

package
com.aplikasi.apliaksiternaktelur.admin.pak
an;

import
androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.Button;

```

```

import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;

import com.aplikasi.apliaksiternaktelur.R;
import com.google.firebase.FirebaseApp;
import
com.google.firebase.database.DatabaseRef
erence;
import
com.google.firebase.database.FirebaseData
base;

```

```

public class Edit_pakan extends
AppCompatActivity {

```

```

    private DatePicker datePickerTanggal;
    private EditText editTextNamaPakan,
editTextHargaPakan,
editTextJumlahPakan;
    private Spinner spinnerJenisPakan;
    private Button btnUpdate, btnHapus,
btnCancel;

```

```

    // Referensi ke Firebase Database
    private DatabaseReference
databaseReference;

```

```

    // Variabel untuk menyimpan data awal
    private String idPakan, nama,
jenisPakan, tanggalPemberian;
    private int hargaPerKg, jumlahPakan;

```

```

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

    setContentView(R.layout.edit_pakan);

```

```

    // Inisialisasi Firebase
    FirebaseApp.initializeApp(this);
    databaseReference =
FirebaseDatabase.getInstance().getReferen
ce().child("admin/pakan");

```

```

        // Ambil data dari Intent
        Intent intent = getIntent();
        idPakan =
intent.getStringExtra("idPakan");
        nama =
intent.getStringExtra("nama");
        hargaPerKg =
intent.getIntExtra("hargaPerKg", 0);
        jumlahPakan =
intent.getIntExtra("jumlahPakan", 0);
        tanggalPemberian =
intent.getStringExtra("tanggalPemberian")
;
        jenisPakan =
intent.getStringExtra("jenisPakan");

```

```

        // Ambil referensi dari layout
        datePickerTanggal =
findViewById(R.id.datePickerTanggal);
        editTextNamaPakan =
findViewById(R.id.editTextNamaPakan);
        editTextHargaPakan =
findViewById(R.id.editTextHargaPakan);
        editTextJumlahPakan =
findViewById(R.id.editTextJumlahPakan);
        spinnerJenisPakan =
findViewById(R.id.spinnerJenisPakan);
        btnUpdate =
findViewById(R.id.btnupdate);
        btnHapus =
findViewById(R.id.btnhapus);
        btnCancel =
findViewById(R.id.btncancel);

```

```

        // Set data ke view
        editTextNamaPakan.setText(nama);

```

```

        editTextHargaPakan.setText(String.valueO
f(hargaPerKg));

```

```

        editTextJumlahPakan.setText(String.valueO
f(jumlahPakan));

```

```

        // Format and set the date
        if (tanggalPemberian != null) {

```

```

        String[] parts =
tanggalPemberian.split("-");
        int year =
Integer.parseInt(parts[0]);
        int month =
Integer.parseInt(parts[1]) - 1;
        int day = Integer.parseInt(parts[2]);

datePickerTanggal.updateDate(year,
month, day);
    }

    // Set spinner value
    ArrayAdapter<CharSequence>
adapter =
ArrayAdapter.createFromResource(this,
        R.array.jenis_pakan_array,
        android.R.layout.simple_spinner_item);

adapter.setDropDownViewResource(android.
id.R.layout.simple_spinner_dropdown_ite
m);

spinnerJenisPakan.setAdapter(adapter);
    if (jenisPakan != null) {
        int spinnerPosition =
adapter.getPosition(jenisPakan);

spinnerJenisPakan.setSelection(spinnerPos
ition);
    }

    // Set OnClickListener for the buttons
    btnHapus.setOnClickListener(v ->
deleteDataPakan());
    btnCancel.setOnClickListener(v ->
finish());
    btnUpdate.setOnClickListener(v ->
updateDataPakan());
}

private void updateDataPakan() {
    // Ambil nilai dari input user
    String namaPakan =
editTextNamaPakan.getText().toString().tri
m();

```

```

        String jumlahPakanStr =
editTextJumlahPakan.getText().toString().t
rim();
        String hargaPakanStr =
editTextHargaPakan.getText().toString().tri
m();
        String jenisPakan =
spinnerJenisPakan.getSelectedItem().toStri
ng();

        // Validasi apakah ada input yang
kosong
        if (namaPakan.isEmpty() ||
jumlahPakanStr.isEmpty() ||
hargaPakanStr.isEmpty() ||
jenisPakan.equals("Pilih")) {
            Toast.makeText(this, "Harap
lengkapi semua data",
Toast.LENGTH_SHORT).show();
            return;
        }

        // Ambil nilai jumlahPakan dan
hargaPakan setelah validasi
        int jumlahPakan =
Integer.parseInt(jumlahPakanStr);
        int hargaPakan =
Integer.parseInt(hargaPakanStr);

        // Ambil tanggal dari DatePicker
        int day =
datePickerTanggal.getDayOfMonth();
        int month =
datePickerTanggal.getMonth() + 1; //
DatePicker month is zero-based
        int year =
datePickerTanggal.getYear();
        String tanggalPemberian = year + "-"
+ month + "-" + day;

        // Buat objek DataPakan yang
diperbarui
        DataPakan dataPakan = new
DataPakan(idPakan, namaPakan,
hargaPakan, jumlahPakan, "",
tanggalPemberian, jenisPakan);

```

```

        // Update data ke Firebase
        databaseReference.child(idPakan).setValue
        (dataPakan)
            .addOnSuccessListener(aVoid ->
        {

            Toast.makeText(Edit_pakan.this, "Data
            pakan berhasil diperbarui",
            Toast.LENGTH_SHORT).show();
                finish();
                Intent intent = new
            Intent(Edit_pakan.this,
            PakanActivity.class);
                startActivity(intent);
            })
            .addOnFailureListener(e -> {

            Toast.makeText(Edit_pakan.this, "Gagal
            memperbarui data: " + e.getMessage(),
            Toast.LENGTH_SHORT).show();
                });
        }

        private void deleteDataPakan() {

        databaseReference.child(idPakan).remove
        Value()
            .addOnSuccessListener(aVoid ->
        {

            Toast.makeText(Edit_pakan.this, "Data
            pakan berhasil dihapus",
            Toast.LENGTH_SHORT).show();
                finish();
                Intent intent = new
            Intent(Edit_pakan.this,
            PakanActivity.class);
                startActivity(intent);
            })
            .addOnFailureListener(e -> {

            Toast.makeText(Edit_pakan.this, "Gagal
            menghapus data: " + e.getMessage(),
            Toast.LENGTH_SHORT).show();

```

```

        });
    }
}

```

PakanActivity.java

```

package
com.aplikasi.apliaksiternaktelur.admin.pak
an;

import android.content.Intent;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;
import
androidx.recyclerview.widget.LinearLayou
tManager;
import
androidx.recyclerview.widget.RecyclerVie
w;

import
com.aplikasi.apliaksiternaktelur.MainActiv
ity;
import com.aplikasi.apliaksiternaktelur.R;
import
com.google.firebase.database.DataSnapsho
t;
import
com.google.firebase.database.DatabaseErr
or;
import
com.google.firebase.database.DatabaseRef
erence;
import
com.google.firebase.database.FirebaseData
base;

```

```

import
com.google.firebase.database.ValueEventListener;

import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

public class PakanActivity extends
AppCompatActivity {

    private RecyclerView
recyclerViewPakan;
    private PakanAdapter adapter;
    private List<DataPakan> dataPakanList;
    private List<DataPakan> filteredList; //
Untuk menyimpan hasil filter

    private TextView txtTotalPengeluaran;
    private EditText txtCari;

    private DatabaseReference adminRef;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.data_pakan);

        txtTotalPengeluaran =
findViewById(R.id.txtJumlahPengeluaran
);
        recyclerViewPakan =
findViewById(R.id.recyclerViewPakan);
        txtCari = findViewById(R.id.txtCari);
        // EditText untuk pencarian

        recyclerViewPakan.setLayoutManager(new
LinearLayoutManager(this));

        dataPakanList = new ArrayList<>();
        filteredList = new ArrayList<>(); //
Inisialisasi filteredList

```

```

        adapter = new
PakanAdapter(dataPakanList);

        recyclerViewPakan.setAdapter(adapter);

        FirebaseDatabase database =
FirebaseDatabase.getInstance();
        adminRef =
database.getReference("admin/pakan");

        fetchDataPakan();

        // Set listener untuk EditText
pencarian
        txtCari.addTextChangedListener(new
TextWatcher() {
            @Override
            public void
beforeTextChanged(CharSequence s, int
start, int count, int after) {
                // Do nothing
            }

            @Override
            public void
onTextChanged(CharSequence s, int start,
int before, int count) {
                // Filter list based on input text
                filterData(s.toString());
            }

            @Override
            public void
afterTextChanged(Editable s) {
                // Do nothing
            }
        });

        private void fetchDataPakan() {

            adminRef.addValueEventListener(new
ValueEventListener() {
                @Override
                public void
onDataChange(@NonNull DataSnapshot

```

```

dataSnapshot) {
    dataPakanList.clear();

    for (DataSnapshot snapshot :
dataSnapshot.getChildren()) {
        String id =
snapshot.child("id").getValue(String.class)
;
        String nama =
snapshot.child("nama").getValue(String.cl
ass);
        Integer hargaPerKgInteger =
snapshot.child("hargaPerKg").getValue(Int
eger.class);
        int hargaPerKg =
hargaPerKgInteger != null ?
hargaPerKgInteger.intValue() : 0; // Handle
null case
        int jumlahPakan =
snapshot.child("jumlahPakan").getValue(I
nteger.class);
        String keteranganPemberian =
snapshot.child("keteranganPemberian").ge
tValue(String.class);
        String tanggalPemberian =
snapshot.child("tanggalPemberian").getVal
ue(String.class);
        String jenisPakan =
snapshot.child("jenisPakan").getValue(Stri
ng.class);
        DataPakan dataPakan = new
DataPakan(id, nama, hargaPerKg,
jumlahPakan,
keteranganPemberian,tanggalPemberian,
jenisPakan);

dataPakanList.add(dataPakan);
    }

    adapter.notifyDataSetChanged();
    calculateTotalPengeluaran(); //
Ensure this method is called after updating
dataPakanList
}

@Override

```

```

    public void
onCancelled(@NonNull DatabaseError
databaseError) {
        Log.e("Firebase", "Gagal
membaca data pakan",
databaseError.toException());
    }
});
}

    private void calculateTotalPengeluaran()
{
        int totalPengeluaran = 0;

        for (DataPakan data : dataPakanList)
        {
            totalPengeluaran +=
data.getJumlahPakan() *
data.getHargaPerKg();
        }

        Locale localeID = new Locale("in",
"ID"); // Using Indonesian locale
        NumberFormat formatRupiah =
NumberFormat.getCurrencyInstance(local
eID);
        String formattedTotalPengeluaran =
formatRupiah.format(totalPengeluaran);

txtTotalPengeluaran.setText(formattedTota
lPengeluaran);
    }

    private void filterData(String query) {
        filteredList.clear();
        for (DataPakan data : dataPakanList)
        {
            if
(data.getNama().toLowerCase().contains(q
uery.toLowerCase())) {
                filteredList.add(data);
            }
        }
        adapter.filterList(filteredList);
    }
}

```

```

    }

    public void tambah_pakan(View view) {
        Intent intent = new Intent(this,
        tambah_pakan.class);
        startActivity(intent);
        finish();
    }

    public void pressback (View view) {
        Intent intent = new Intent(this,
        MainActivity.class);
        startActivity(intent);
        finish();
    }
}

```

PakanAdapter

```

package
com.aplikasi.apliaksiternaktelur.admin.pakan;

import android.content.Intent;
import android.os.Bundle;
import android.textEditable;
import android.text.TextWatcher;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;
import
androidx.recyclerview.widget.LinearLayoutManager;
import
androidx.recyclerview.widget.RecyclerView;

import
com.aplikasi.apliaksiternaktelur.MainActivity;
import com.aplikasi.apliaksiternaktelur.R;

```

```

import
com.google.firebase.database.DataSnapshot;
import
com.google.firebase.database.DatabaseError;
import
com.google.firebase.database.DatabaseReference;
import
com.google.firebase.database.FirebaseDatabase;
import
com.google.firebase.database.ValueEventListener;

```

```

import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

```

```

public class PakanActivity extends
AppCompatActivity {

```

```

    private RecyclerView
recyclerViewPakan;
    private PakanAdapter adapter;
    private List<DataPakan> dataPakanList;
    private List<DataPakan> filteredList; //
Untuk menyimpan hasil filter

```

```

    private TextView txtTotalPengeluaran;
    private EditText txtCari;

```

```

    private DatabaseReference adminRef;

```

```

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

    setContentView(R.layout.data_pakan);

```

```

        txtTotalPengeluaran =
findViewById(R.id.txtJumlahPengeluaran
);

```

```

        recyclerViewPakan =
findViewById(R.id.recyclerViewPakan);
        txtCari = findViewById(R.id.txtCari);
// EditText untuk pencarian

```

```

recyclerViewPakan.setLayoutManager(new
LinearLayoutManager(this));

```

```

        dataPakanList = new ArrayList<>();
        filteredList = new ArrayList<>(); //
Inisialisasi filteredList

```

```

        adapter = new
PakanAdapter(dataPakanList);

```

```

recyclerViewPakan.setAdapter(adapter);

```

```

        FirebaseDatabase database =
FirebaseDatabase.getInstance();
        adminRef =
database.getReference("admin/pakan");

```

```

        fetchDataPakan();

```

```

        // Set listener untuk EditText
pencarian
        txtCari.addTextChangedListener(new
TextWatcher() {
            @Override
            public void
beforeTextChanged(CharSequence s, int
start, int count, int after) {
                // Do nothing
            }

```

```

            @Override
            public void
onTextChanged(CharSequence s, int start,
int before, int count) {
                // Filter list based on input text
                filterData(s.toString());
            }

```

```

            @Override
            public void
afterTextChanged(Editable s) {

```

```

                // Do nothing
            }
        });
    }

```

```

        private void fetchDataPakan() {

```

```

            adminRef.addValueEventListener(new
ValueEventListener() {

```

```

                @Override
                public void
onDataChange(@NonNull DataSnapshot
dataSnapshot) {
                    dataPakanList.clear();

```

```

                        for (DataSnapshot snapshot :
dataSnapshot.getChildren()) {
                            String id =
snapshot.child("id").getValue(String.class)
;

```

```

                                String nama =
snapshot.child("nama").getValue(String.cl
ass);

```

```

                                    Integer hargaPerKgInteger =
snapshot.child("hargaPerKg").getValue(Int
eger.class);

```

```

                                        int hargaPerKg =
hargaPerKgInteger != null ?
hargaPerKgInteger.intValue() : 0; // Handle
null case

```

```

                                            int jumlahPakan =
snapshot.child("jumlahPakan").getValue(I
nteger.class);

```

```

                                                String keteranganPemberian =
snapshot.child("keteranganPemberian").ge
tValue(String.class);

```

```

                                                    String tanggalPemberian =
snapshot.child("tanggalPemberian").getVal
ue(String.class);

```

```

                                                        String jenisPakan =
snapshot.child("jenisPakan").getValue(Stri
ng.class);

```

```

                                                            DataPakan dataPakan = new
DataPakan(id, nama, hargaPerKg,
jumlahPakan,
keteranganPemberian,tanggalPemberian,

```

```

jenisPakan);

dataPakanList.add(dataPakan);
    }

    adapter.notifyDataSetChanged();
    calculateTotalPengeluaran(); //
Ensure this method is called after updating
dataPakanList
    }

    @Override
    public void
onCancelled(@NonNull DatabaseError
databaseError) {
        Log.e("Firebase", "Gagal
membaca data pakan",
databaseError.toException());
    }
    });
}

private void calculateTotalPengeluaran()
{
    int totalPengeluaran = 0;

    for (DataPakan data : dataPakanList)
    {
        totalPengeluaran +=
data.getJumlahPakan() *
data.getHargaPerKg();
    }

    Locale localeID = new Locale("in",
"ID"); // Using Indonesian locale
    NumberFormat formatRupiah =
NumberFormat.getCurrencyInstance(localeID);
    String formattedTotalPengeluaran =
formatRupiah.format(totalPengeluaran);

    txtTotalPengeluaran.setText(formattedTotalPengeluaran);
}

```

```

private void filterData(String query) {
    filteredList.clear();
    for (DataPakan data : dataPakanList)
    {
        if
(data.getNama().toLowerCase().contains(query.toLowerCase())) {
            filteredList.add(data);
        }
    }
    adapter.filterList(filteredList);
}

public void tambah_pakan(View view) {
    Intent intent = new Intent(this,
tambah_pakan.class);
    startActivity(intent);
    finish();
}

public void pressback (View view) {
    Intent intent = new Intent(this,
MainActivity.class);
    startActivity(intent);
    finish();
}
}

```

tambah_pakan.java

```

package
com.aplikasi.apliksiternaktelur.admin.pakan;

import android.app.DatePickerDialog;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;

```

```

import android.widget.Toast;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;

import
com.aplikasi.apliaksiternaktelur.MainActivity;
import com.aplikasi.apliaksiternaktelur.R;
import
com.aplikasi.apliaksiternaktelur.admin.ayam.AyamActivity;
import
com.aplikasi.apliaksiternaktelur.admin.ayam.TambahAyam;
import
com.aplikasi.apliaksiternaktelur.admin.obat.tambah_obat;
import
com.google.android.gms.tasks.OnFailureListener;
import
com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.FirebaseApp;
import
com.google.firebase.database.DataSnapshot;
import
com.google.firebase.database.DatabaseReference;
import
com.google.firebase.database.FirebaseDatabase;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;

public class tambah_pakan extends
AppCompatActivity {

    private EditText editTextNamaPakan,
    editTextJumlahPakan,

```

```

    editTextHargaPakan;
    private Spinner spinnerJenisPakan;
    private DatePicker datePicker;
    private Button buttonTambah;
    private TextView tvDatePicker;
    // Referensi ke Firebase Database
    private DatabaseReference
    databaseReference,totalHargaPakan;

    // Array untuk harga pakan
    private String[] hargaPakanArray;

    @Override
    protected void onCreate(Bundle
    savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_tambah_
        pakan);

        // Inisialisasi Firebase
        FirebaseApp.initializeApp(this);

        // Inisialisasi views
        editTextNamaPakan =
        findViewById(R.id.editTextNamaPakan);
        editTextJumlahPakan =
        findViewById(R.id.editTextJumlahPakan);
        editTextHargaPakan =
        findViewById(R.id.editTextHargaPakan);
        spinnerJenisPakan =
        findViewById(R.id.spinnerJenisPakan);
        tvDatePicker =
        findViewById(R.id.tvDatePickerPakan);
        buttonTambah =
        findViewById(R.id.buttonTambah);

        // Inisialisasi Firebase Database
        databaseReference =
        FirebaseDatabase.getInstance().getReference()
        .child("admin/pakan");
        totalHargaPakan =
        FirebaseDatabase.getInstance().getReference()
        .child("admin/totalHargaPakan");

        // Ambil array harga pakan dari

```

```

resources
    hargaPakanArray =
    getResources().getStringArray(R.array.har
    ga_pakan_array);

    // Setup ArrayAdapter untuk Spinner
    jenis pakan
    ArrayAdapter<CharSequence>
    adapter =
    ArrayAdapter.createFromResource(this,
        R.array.jenis_pakan_array,
        android.R.layout.simple_spinner_item);

    adapter.setDropDownViewResource(andro
    id.R.layout.simple_spinner_dropdown_ite
    m);

    spinnerJenisPakan.setAdapter(adapter);

    // Set listener untuk Spinner jenis
    pakan

    spinnerJenisPakan.setOnItemSelectedListener(new
    AdapterView.OnItemSelectedListener() {
        @Override
        public void
        onItemSelected(AdapterView<?> parent,
        View view, int position, long id) {
            // Saat pilihan jenis pakan
            berubah, set harga pakan sesuai dengan
            posisi
            if (position >= 0 && position <
            hargaPakanArray.length) {
                String hargaPakanString =
                hargaPakanArray[position];
                int hargaPakan =
                Integer.parseInt(hargaPakanString);

                editTextHargaPakan.setText(String.valueOf(
                f(hargaPakan)));
            }
        }

        @Override
        public void

```

```

onNothingSelected(AdapterView<?>
parent) {
    // Do nothing
}
});

// Setup button click listener

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

SimpleDateFormat sdf = new
SimpleDateFormat("dd-MM-yyyy",
Locale.getDefault());
String currentDate =
sdf.format(Calendar.getInstance().getTime
());
tvDatePicker.setText(currentDate);

tvDatePicker.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Calendar calendar =
        Calendar.getInstance();
        int year =
        calendar.get(Calendar.YEAR);
        int month =
        calendar.get(Calendar.MONTH);
        int day =
        calendar.get(Calendar.DAY_OF_MONTH);

        DatePickerDialog
        datePickerDialog = new
        DatePickerDialog(tambah_pakan.this,
            new
            DatePickerDialog.OnDateSetListener() {
                @Override
                public void
                onDateSet(DatePicker view, int year, int
                month, int dayOfMonth) {

```

```

        Calendar selectedDate
= Calendar.getInstance();
        selectedDate.set(year,
month, dayOfMonth);
        String date =
sdf.format(selectedDate.getTime());

tvDatePicker.setText(date);
    }
    }, year, month, day);
    datePickerDialog.show();
    }
});
}

private void tambahDataPakan() {
    // Ambil nilai dari input user
    String namaPakan =
editTextNamaPakan.getText().toString().trim();
    String jumlahPakanStr =
editTextJumlahPakan.getText().toString().trim();
    String hargaPakanStr =
editTextHargaPakan.getText().toString().trim();
    String jenisPakan =
spinnerJenisPakan.getSelectedItem().toString();
    String tanggalMasukPakan =
tvDatePicker.getText().toString().trim();
    // Validasi apakah ada input yang
kosong
    if (namaPakan.isEmpty() ||
jumlahPakanStr.isEmpty() ||
hargaPakanStr.isEmpty() ||
jenisPakan.equals("Pilih")) {
        Toast.makeText(this, "Harap
lengkapi semua data",
Toast.LENGTH_SHORT).show();
        return;
    }

    // Ambil nilai jumlahPakan dan
hargaPakan setelah validasi
    int jumlahPakan =

```

```

Integer.parseInt(jumlahPakanStr);
    int hargaPakan =
Integer.parseInt(hargaPakanStr);

    // Buat objek DataPakan
    String idPakan =
databaseReference.push().getKey(); //
Generate unique key
    DataPakan dataPakan = new
DataPakan(idPakan, namaPakan,
hargaPakan, jumlahPakan, "",
tanggalMasukPakan, jenisPakan); // Ubah
keteranganPemberian menjadi string
kosong ("" ) karena tidak ada input untuk
itu saat tambah data

    // Simpan data ke Firebase

databaseReference.child(idPakan).setValue
(dataPakan)
        .addOnSuccessListener(new
OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void
aVoid) {

                Toast.makeText(tambah_pakan.this, "Data
pakan berhasil disimpan",
                Toast.LENGTH_SHORT).show();
                // Reset form setelah data
disimpan (opsional)
                resetForm();
                Intent intent = new
Intent(tambah_pakan.this,
PakanActivity.class);
                startActivity(intent);
                finish();
            }
        })
        .addOnFailureListener(new
OnFailureListener() {
            @Override
            public void
onFailure(@NonNull Exception e) {

```

```

Toast.makeText(tambah_pakan.this, "Gagal
menyimpan data: " + e.getMessage(),
Toast.LENGTH_SHORT).show();
    }
});
}

```

```

private void resetForm() {
    editTextNamaPakan.setText("");
    editTextJumlahPakan.setText("");
    editTextHargaPakan.setText("");
    spinnerJenisPakan.setSelection(0); //
Reset Spinner selection
    tvDatePicker.setText("");
}

public void pressback(View view) {
    Intent intent = new
Intent(tambah_pakan.this,
PakanActivity.class);
    startActivity(intent);
    finish();
}
}

```

DataPenjualanPerTanggal.java

```

package
com.aplikasi.apliksiternaktelur.admin.pen
jualan;

```

```

public class DataPenjualanPerTanggal {
    private String tanggal;
    private int jumlahTelurBaik;
    private int totalHargaBaik;
    private int jumlahTelurPecah;
    private int totalHargaPecah;

    public DataPenjualanPerTanggal(String
tanggal) {

```

```

        this.tanggal = tanggal;
        this.jumlahTelurBaik = 0;
        this.totalHargaBaik = 0;
        this.jumlahTelurPecah = 0;
        this.totalHargaPecah = 0;
    }

```

```

    public String getTanggal() {
        return tanggal;
    }

```

```

    public void setTanggal(String tanggal) {
        this.tanggal = tanggal;
    }

```

```

    public int getJumlahTelurBaik() {
        return jumlahTelurBaik;
    }

```

```

    public void setJumlahTelurBaik(int
jumlahTelurBaik) {
        this.jumlahTelurBaik =
jumlahTelurBaik;
    }

```

```

    public int getTotalHargaBaik() {
        return totalHargaBaik;
    }

```

```

    public void setTotalHargaBaik(int
totalHargaBaik) {
        this.totalHargaBaik = totalHargaBaik;
    }

```

```

    public int getJumlahTelurPecah() {
        return jumlahTelurPecah;
    }

```

```

    public void setJumlahTelurPecah(int
jumlahTelurPecah) {
        this.jumlahTelurPecah =
jumlahTelurPecah;
    }

```

```

    public int getTotalHargaPecah() {
        return totalHargaPecah;
    }

```

```

    }

    public void setTotalHargaPecah(int
totalHargaPecah) {
        this.totalHargaPecah =
totalHargaPecah;
    }

    public int getTotal() {
        return totalHargaBaik +
totalHargaPecah;
    }
}

```

PenjualanActivity.java

```

package
com.aplikasi.apliaksiternaktelur.admin.pen
jualan;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;
import
androidx.recyclerview.widget.LinearLayou
tManager;
import
androidx.recyclerview.widget.RecyclerVie
w;

import
com.aplikasi.apliaksiternaktelur.MainActiv
ity;
import com.aplikasi.apliaksiternaktelur.R;
import
com.aplikasi.apliaksiternaktelur.admin.telur.table_telur;
import

```

```

com.google.android.material.floatingaction
button.FloatingActionButton;
import
com.google.firebase.database.DataSnapsho
t;
import
com.google.firebase.database.DatabaseErr
or;
import
com.google.firebase.database.DatabaseRef
erence;
import
com.google.firebase.database.FirebaseData
base;
import
com.google.firebase.database.ValueEventL
istener;

```

```

import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

```

```

public class PenjualanActivity extends
AppCompatActivity {

```

```

    private static final String TAG =
"PenjualanActivity";

```

```

    private RecyclerView recyclerView;
    private PenjualanAdapter adapter;
    private List<PenjualanData>
penjualanList;

```

```

    private TextView tvTotalPemasukan;
    private ImageView ivBack;
    private FloatingActionButton
btnTambahData;

```

```

    private DatabaseReference
databaseReference;

```

```

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```
setContentView(R.layout.activity_penjuala
n);
```

```
    // Inisialisasi UI
    recyclerView =
findViewById(R.id.recyclerView);
    tvTotalPemasukan =
findViewById(R.id.tvTotalPemasukan);
    ivBack = findViewById(R.id.ivBack);
    btnTambahData =
findViewById(R.id.btnTambahData);
```

```
    // Inisialisasi RecyclerView
    recyclerView.setLayoutManager(new
LinearLayoutManager(this));
    penjualanList = new ArrayList<>();
    adapter = new PenjualanAdapter(this,
penjualanList);
    recyclerView.setAdapter(adapter);
```

```
    // Inisialisasi Firebase
    databaseReference =
FirebaseDatabase.getInstance().getReferen
ce().child("admin").child("pemasukan");
```

```
    // Mendapatkan data penjualan dari
Firebase
    fetchPenjualanData();
```

```
    // Listener untuk tombol kembali
    ivBack.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new
Intent(PenjualanActivity.this,
MainActivity.class);
            startActivity(intent);
            finish();
        }
    });
```

```
    // Listener untuk tombol tambah data
    btnTambahData.setOnClickListener(new
```

```
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // Navigasi ke activity untuk
menambah data penjualan
            Intent intent = new
Intent(PenjualanActivity.this,
Tambah_penjualan.class);
            startActivity(intent);
            finish();
        }
    });
}
```

```
    // Method untuk mengambil data
penjualan dari Firebase
    private void fetchPenjualanData() {
```

```
        databaseReference.addValueEventListener
(new ValueEventListener() {
            @Override
            public void
onDataChange(@NonNull DataSnapshot
snapshot) {
                penjualanList.clear();
                int totalPemasukan = 0;
```

```
                for (DataSnapshot dataSnapshot
: snapshot.getChildren()) {
                    PenjualanData penjualan =
dataSnapshot.getValue(PenjualanData.clas
s);
                    penjualanList.add(penjualan);
```

```
                // Menghitung total
pemasukan
                totalPemasukan +=
penjualan.getTotalHarga();
            }
        }
```

```
        NumberFormat formatter =
NumberFormat.getCurrencyInstance(new
Locale("id", "ID")); // Menggunakan
format mata uang Indonesia (IDR)
        String formattedTotal =
formatter.format(totalPemasukan);
```

```

        // Menambahkan titik di depan
        simbol mata uang "Rp"
        String currencySymbol =
        formatter.getCurrency().getSymbol(new
        Locale("id", "ID"));
        formattedTotal = "Rp." +
        formattedTotal.replace(currencySymbol,
        "");

        // Menampilkan jumlah
        pemasukan pada TextView

        tvTotalPemasukan.setText(formattedTotal)
        ;

        // Mengupdate RecyclerView
        adapter.notifyDataSetChanged();
    }

    @Override
    public void
    onCancelled(@NonNull DatabaseError
    error) {
        Log.e(TAG, "Gagal mengambil
        data penjualan: " + error.getMessage());
    }
    });
}

    public void btn_tablepenjualan(View
    view) {
        Intent intent = new Intent(this,
        TablePenjualan.class);

        intent.addFlags(Intent.FLAG_ACTIVITY_
        CLEAR_TOP |
        Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
        finish();
    }
}

```

PenjualanAdapter.java

```

package
com.aplikasi.apliaksiternaktelur.admin.pen
jualan;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import
androidx.recyclerview.widget.RecyclerVie
w;

import com.aplikasi.apliaksiternaktelur.R;
import
com.google.android.gms.tasks.OnFailureL
istener;
import
com.google.android.gms.tasks.OnSuccess
Listener;
import
com.google.firebase.database.DataSnapsho
t;
import
com.google.firebase.database.DatabaseErr
or;
import
com.google.firebase.database.DatabaseRef
erence;
import
com.google.firebase.database.FirebaseData
base;
import
com.google.firebase.database.ValueEventL
istener;

import java.util.List;

public class PenjualanAdapter extends

```

```

RecyclerView.Adapter<PenjualanAdapter.
ViewHolder> {
    private Context context;
    private List<PenjualanData>
penjualanList;

```

```

    public PenjualanAdapter(Context
context, List<PenjualanData>
penjualanList) {
        this.context = context;
        this.penjualanList = penjualanList;
    }

```

```

    @NonNull
    @Override
    public ViewHolder
onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
        View view =
LayoutInflater.from(context).inflate(R.layo
ut.item_penjualan, parent, false);
        return new ViewHolder(view);
    }

```

```

    @Override
    public void
onBindViewHolder(@NonNull
ViewHolder holder, int position) {
        PenjualanData penjualan =
penjualanList.get(position);
        holder.bind(penjualan);
    }

```

```

    @Override
    public int getItemCount() {
        return penjualanList.size();
    }

```

```

    public class ViewHolder extends
RecyclerView.ViewHolder {

```

```

        private TextView
tanggalJualTextView;
        private TextView
jumlahTelurTextView;
        private TextView

```

```

hargaSatuanTextView;
        private TextView
totalHargaTextView;
        private ImageButton deleteIcon;

```

```

        public ViewHolder(@NonNull View
itemView) {
            super(itemView);
            tanggalJualTextView =
itemView.findViewById(R.id.tanggalJualT
extView);
            jumlahTelurTextView =
itemView.findViewById(R.id.jumlahTelur
TextView);
            hargaSatuanTextView =
itemView.findViewById(R.id.hargaSatuan
TextView);
            totalHargaTextView =
itemView.findViewById(R.id.totalHargaT
extView);
            deleteIcon =
itemView.findViewById(R.id.deleteIcon);

```

```

            deleteIcon.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View v) {

showDeleteConfirmationDialog();
                }
            });
        }

```

```

        private void
showDeleteConfirmationDialog() {
            AlertDialog.Builder builder = new
AlertDialog.Builder(context);
            builder.setTitle("Konfirmasi
Hapus");
            builder.setMessage("Apakah Anda
yakin ingin menghapus item ini?");
            builder.setPositiveButton("Ya",
new DialogInterface.OnClickListener() {
                @Override
                public void
onClick(DialogInterface dialog, int which)

```

```

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

private void deleteItem() {
    int position = getAdapterPosition();
    if (position !=
RecyclerView.NO_POSITION) {
        PenjualanData datapenjualan =
penjualanList.get(position);

        // Menghapus data dari Firebase
        String idPenjualan =
datapenjualan.getIdPenjualan();
        DatabaseReference penjualanRef
=
FirebaseDatabase.getInstance().getReferen
ce()

        .child("admin").child("pemasukan").child(i
dPenjualan);

        penjualanRef.removeValue().addOnSucces
sListener(new OnSuccessListener<Void>()
{
            @Override
            public void onSuccess(Void
aVoid) {
                // Data berhasil dihapus dari
                Firebase

```

```

        // Memperbarui total telur di
        'admin/totaltelur'
        DatabaseReference
        totalTelurRef =
        FirebaseDatabase.getInstance().getReferen
        ce()

        .child("admin").child("totaltelur");

        totalTelurRef.addListenerForSingleValueE
        vent(new ValueEventListener() {
            @Override
            public void
            onDataChange(@NonNull DataSnapshot
            dataSnapshot) {
                if
                (dataSnapshot.exists()) {
                    int
                    totalTelurSekarang =
                    dataSnapshot.getValue(Integer.class);

                    // Update total telur
                    int
                    jumlahTelurDihapus =
                    datapenjualan.getJumlahTelur();
                    int totalTelurBaru =
                    totalTelurSekarang + jumlahTelurDihapus;

                    totalTelurRef.setValue(totalTelurBaru)

                    .addOnSuccessListener(new
                    OnSuccessListener<Void>() {
                        @Override
                        public void
                        onSuccess(Void aVoid) {
                            // Update
                            UI or perform other actions if needed
                        }
                    })

                    .addOnFailureListener(new
                    OnFailureListener() {
                        @Override
                        public void

```

```

onFailure(@NonNull Exception e) {
    // Handle
    failure
    }
    });
}

@Override
public void
onCancelled(@NonNull DatabaseError
databaseError) {
    // Handle error
    }
    });
}
}).addOnFailureListener(new
OnFailureListener() {
    @Override
    public void
onFailure(@NonNull Exception e) {
    // Handle failure
    }
    });
}

    public void bind(PenjualanData
penjualan) {

tanggalJualTextView.setText(penjualan.get
TanggalPenjualan());

jumlahTelurTextView.setText(String.value
Of(penjualan.getJumlahTelur()));
    hargaSatuanTextView.setText("Rp.
" + penjualan.getHargaSatuan());
    totalHargaTextView.setText("Rp. "
+ penjualan.getTotalHarga());
    }
}
}

```

PenjualanData.java

```

package
com.aplikasi.apliksiternaktelur.admin.pen
jualan;
public class PenjualanData {
    private String idPenjualan;
    private int jumlahTelur;
    private int hargaSatuan;
    private int totalHarga;
    private String tanggalPenjualan;
    private String jenisTelur;

    public PenjualanData() {
        // Default constructor required for
calls to
DataSnapshot.getValue(PenjualanData.clas
s)
    }

    public PenjualanData(String
idPenjualan, int jumlahTelur, int
hargaSatuan, int totalHarga, String
tanggalPenjualan, String jenisTelur) {
        this.idPenjualan = idPenjualan;
        this.jumlahTelur = jumlahTelur;
        this.hargaSatuan = hargaSatuan;
        this.totalHarga = totalHarga;
        this.tanggalPenjualan =
tanggalPenjualan;
        this.jenisTelur = jenisTelur;
    }

    // Getter and setter methods for all
properties

    public String getIdPenjualan() {
        return idPenjualan;
    }

    public void setIdPenjualan(String
idPenjualan) {
        this.idPenjualan = idPenjualan;
    }

    public int getJumlahTelur() {
        return jumlahTelur;
    }
}

```

```

    public void setJumlahTelur(int
jumlahTelur) {
        this.jumlahTelur = jumlahTelur;
    }

    public int getHargaSatuan() {
        return hargaSatuan;
    }

    public void setHargaSatuan(int
hargaSatuan) {
        this.hargaSatuan = hargaSatuan;
    }

    public int getTotalHarga() {
        return totalHarga;
    }

    public void setTotalHarga(int
totalHarga) {
        this.totalHarga = totalHarga;
    }

    public String getTanggalPenjualan() {
        return tanggalPenjualan;
    }

    public void setTanggalPenjualan(String
tanggalPenjualan) {
        this.tanggalPenjualan =
tanggalPenjualan;
    }

    public String getJenisTelur() {
        return jenisTelur;
    }

    public void setJenisTelur(String
jenisTelur) {
        this.jenisTelur = jenisTelur;
    }
}

```

TablePenjualan.java

```

package
com.aplikasi.apliksiternaktelur.admin.pen
jualan;

import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.Gravity;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;

import com.aplikasi.apliksiternaktelur.R;
import
com.google.firebase.database.DataSnapsho
t;
import
com.google.firebase.database.DatabaseErr
or;
import
com.google.firebase.database.DatabaseRef
erence;
import
com.google.firebase.database.FirebaseData
base;
import
com.google.firebase.database.ValueEventL
istener;

import java.util.HashMap;

public class TablePenjualan extends
AppCompatActivity {

    private TableLayout tableLayout;
    private DatabaseReference
pemasukanRef;

    @Override

```

```

        protected void onCreate(Bundle
savedInstanceState) {
            super.onCreate(savedInstanceState);

setContentView(R.layout.activity_table_pe
njualan); // Ensure you have this layout

            tableLayout =
findViewById(R.id.tableLayout);
            pemasukanRef =
FirebaseDatabase.getInstance().getReferen
ce("admin/pemasukan");

            fetchData();
        }

        private void fetchData() {

pemasukanRef.addValueEventListener(ne
w ValueEventListener() {
            @Override
            public void
onDataChange(@NonNull DataSnapshot
dataSnapshot) {
                HashMap<String,
DataPenjualanPerTanggal> penjualanMap
= new HashMap<>();

                for (DataSnapshot snapshot :
dataSnapshot.getChildren()) {
                    PenjualanData penjualan =
snapshot.getValue(PenjualanData.class);
                    if (penjualan != null) {
                        String tanggal =
penjualan.getTanggalPenjualan();
                        DataPenjualanPerTanggal
dailyData =
penjualanMap.getDefault(tanggal, new
DataPenjualanPerTanggal(tanggal));

                        if
(penjualan.getJenisTelur().equals("baik"))
{

dailyData.setJumlahTelurBaik(dailyData.g
etJumlahTelurBaik() +

```

```

penjualan.getJumlahTelur());

dailyData.setTotalHargaBaik(dailyData.get
TotalHargaBaik() +
penjualan.getTotalHarga());
                } else if
(penjualan.getJenisTelur().equals("pecah")
) {

dailyData.setJumlahTelurPecah(dailyData.
getJumlahTelurPecah() +
penjualan.getJumlahTelur());

dailyData.setTotalHargaPecah(dailyData.g
etTotalHargaPecah() +
penjualan.getTotalHarga());
                }

                penjualanMap.put(tanggal,
dailyData);
            }
        }

        displayData(penjualanMap);
    }

    @Override
    public void
onCancelled(@NonNull DatabaseError
databaseError) {

Toast.makeText(TablePenjualan.this,
"Failed to load data.",
Toast.LENGTH_SHORT).show();
    }
    });
}

    private void
displayData(HashMap<String,
DataPenjualanPerTanggal> penjualanMap)
{
        // Clear previous data
        tableLayout.removeAllViews();

        // Create headers

```

```

        TableRow headerRow = new
        TableRow(this);

        headerRow.addView(createTextView("Tan
        ggal \n", true));

        headerRow.addView(createTextView("Tel
        ur \n Baik", true));

        headerRow.addView(createTextView("Tot
        al \n Baik", true));

        headerRow.addView(createTextView("Tel
        ur \n Pecah", true));

        headerRow.addView(createTextView("Tot
        al \n Pecah", true));

        headerRow.addView(createTextView("Tot
        al \n", true));
        tableLayout.addView(headerRow);

        // Populate rows
        for (DataPenjualanPerTanggal data :
        penjualanMap.values()) {
            TableRow row = new
            TableRow(this);

            row.addView(createTextView(data.getTan
            ggal()));

            row.addView(createTextView(String.value
            Of(data.getJumlahTelurBaik())));

            row.addView(createTextView(String.value
            Of(data.getTotalHargaBaik())));

            row.addView(createTextView(String.value
            Of(data.getJumlahTelurPecah())));

            row.addView(createTextView(String.value
            Of(data.getTotalHargaPecah())));

            row.addView(createTextView(String.value
            Of(data.getTotal())));
            tableLayout.addView(row);

```

```

        }
    }

    private TextView createTextView(String
    text) {
        return createTextView(text, false);
    }

    private TextView createTextView(String
    text, boolean isHeader) {
        TextView textView = new
        TextView(this);
        textView.setText(text);
        textView.setPadding(8, 8, 8, 8);

        textView.setGravity(Gravity.CENTER);

        // Styling for headers
        if (isHeader) {
            textView.setTypeface(null,
            Typeface.BOLD);

            textView.setBackgroundColor(Color.LTGR
            AY);
        }

        // Set layout parameters for equal
        column widths
        TableRow.LayoutParams params =
        new TableRow.LayoutParams(

        TableRow.LayoutParams.WRAP_CONTE
        NT,

        TableRow.LayoutParams.WRAP_CONTE
        NT,

        1.0f // Weight to distribute space
        equally
        );
        textView.setLayoutParams(params);

        return textView;
    }
}

```

Tambah_penjualan.java

```
package
com.aplikasi.apliaksiternaktelur.admin.pen
jualan;

import android.annotation.SuppressLint;
import android.app.DatePickerDialog;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;

import
com.aplikasi.apliaksiternaktelur.MainActivity;
import com.aplikasi.apliaksiternaktelur.R;
import
com.aplikasi.apliaksiternaktelur.admin.telur.TambahTelur;
import
com.google.android.gms.tasks.OnFailureListener;
import
com.google.android.gms.tasks.OnSuccessListener;
import
com.google.firebase.database.DataSnapshot;
import
com.google.firebase.database.DatabaseError;
import
com.google.firebase.database.DatabaseReference;
```

```
com.google.firebase.database.FirebaseData
base;
import
com.google.firebase.database.ValueEventL
istener;
```

```
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;
```

```
public class Tambah_penjualan extends
AppCompatActivity {
```

```
    private EditText etJumlahTelur,
etHargaSatuan;
    private TextView tvTotalTelur;
    private DatePicker datePicker;
    private Button btnSimpan;
    private RadioGroup
radioGroupJenisTelur;
    private RadioButton radioBaik,
radioPecah;
    private RadioButton radioHancur;
    private TextView tvDatePicker;
    // Referensi ke Firebase Database
    private DatabaseReference
pemasukanRef;
    private DatabaseReference
totalTelurRef;
```

```
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_tambah_
penjualan);
```

```
        // Inisialisasi views
        etJumlahTelur =
findViewById(R.id.etJumlahTelur);
        etHargaSatuan =
findViewById(R.id.etHargaSatuan);
        tvTotalTelur =
findViewById(R.id.tvTotalTelur);
        btnSimpan =
```

```

findViewById(R.id.btnSimpan);
    radioGroupJenisTelur =
findViewById(R.id.radioGroupJenisTelur)
;
    radioBaik =
findViewById(R.id.radioBaik);
    radioPecah =
findViewById(R.id.radioPecah);
    tvDatePicker =
findViewById(R.id.tvDatePickerPenjuala
n);
    // Inisialisasi Firebase Database
    pemasukanRef =
FirebaseDatabase.getInstance().getReferen
ce().child("admin").child("pemasukan");
    totalTelurRef =
FirebaseDatabase.getInstance().getReferen
ce().child("admin").child("total_telur");

    // Setup radio button change listener

radioGroupJenisTelur.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener()
{
    @Override
    public void
onCheckedChanged(RadioGroup group,
int checkedId) {
        if (checkedId == R.id.radioBaik)
        {
            fetchTotalTelur("baik"); //
Fetch total telur baik when selected
        } else if (checkedId ==
R.id.radioPecah) {
            fetchTotalTelur("pecah"); //
Fetch total telur pecah when selected
        }
    }
});
    SimpleDateFormat sdf = new
SimpleDateFormat("dd-MM-yyyy",
Locale.getDefault());
    String currentDate =
sdf.format(Calendar.getInstance().getTime
());

```

```

tvDatePicker.setText(currentDate);

tvDatePicker.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Calendar calendar =
Calendar.getInstance();
        int year =
calendar.get(Calendar.YEAR);
        int month =
calendar.get(Calendar.MONTH);
        int day =
calendar.get(Calendar.DAY_OF_MONTH);

        DatePickerDialog
datePickerDialog = new
DatePickerDialog(Tambah_penjualan.this,
new
DatePickerDialog.OnDateSetListener() {
            @Override
            public void
onDateSet(DatePicker view, int year, int
month, int dayOfMonth) {
                Calendar selectedDate
= Calendar.getInstance();
                selectedDate.set(year,
month, dayOfMonth);
                String date =
sdf.format(selectedDate.getTime());

tvDatePicker.setText(date);
            }
        }, year, month, day);
        datePickerDialog.show();
    }
});

// Setup button click listener
btnSimpan.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        simpanDataPenjualan();
    }
});

```

```

    }

    // Metode untuk mengambil total telur
    dari Firebase
    private void fetchTotalTelur(String
    jenisTelur) {
        DatabaseReference telurRef =
        totalTelurRef.child(jenisTelur);

        telurRef.addListenerForSingleValueEvent(
        new ValueEventListener() {
            @Override
            public void
            onDataChange(@NonNull DataSnapshot
            snapshot) {
                if (snapshot.exists()) {
                    long totalTelur =
                    snapshot.getValue(Long.class);
                    tvTotalTelur.setText("Stok " +
                    jenisTelur + ": " + totalTelur + " Butir");
                } else {
                    tvTotalTelur.setText("Stok
                    telur " + jenisTelur + ": Data tidak
                    tersedia");
                }
            }
        })

        @Override
        public void
        onCancelled(@NonNull DatabaseError
        error) {
            // Handle error
            tvTotalTelur.setText("Stok telur "
            + jenisTelur + ": Data tidak tersedia");
        }
    }

    // Metode untuk menyimpan data
    penjualan ke Firebase
    private void simpanDataPenjualan() {
        // Ambil nilai dari input user
        String jumlahTelurStr =
        etJumlahTelur.getText().toString().trim();

```

```

        String hargaSatuanStr =
        etHargaSatuan.getText().toString().trim();
        String tanggalMasukPenjualan =
        tvDatePicker.getText().toString().trim();
        // Validasi input
        if (jumlahTelurStr.isEmpty() ||
        hargaSatuanStr.isEmpty()) {
            Toast.makeText(this, "Harap
            lengkapi semua data",
            Toast.LENGTH_SHORT).show();
            return;
        }

        // Ambil nilai jumlahTelur dan
        hargaSatuan setelah validasi
        int jumlahTelur =
        Integer.parseInt(jumlahTelurStr);
        int hargaSatuan =
        Integer.parseInt(hargaSatuanStr);

        // Hitung total harga
        int totalHarga = jumlahTelur *
        hargaSatuan;

        // Ambil jenis telur dari RadioGroup
        String jenisTelur =
        radioBaik.isChecked() ? "baik" : "pecah";

        // Referensi ke node yang sesuai
        berdasarkan jenis telur
        DatabaseReference telurRef =
        FirebaseDatabase.getInstance().getReferen
        ce()

        .child("admin").child("total_telur").child(j
        enisTelur);

        // Validasi jumlah telur tidak melebihi
        stok yang tersedia

        telurRef.addListenerForSingleValueEvent(
        new ValueEventListener() {
            @Override
            public void

```

```

onDataChange(@NonNull DataSnapshot
snapshot) {
    if (snapshot.exists()) {
        long totalTelurSekarang =
snapshot.getValue(Long.class);

        // Validasi jumlah telur
        if (jumlahTelur >
totalTelurSekarang) {
            // Jumlah telur melebihi
stok yang tersedia

Toast.makeText(Tambah_penjualan.this,
"Jumlah telur melebihi stok yang tersedia",
Toast.LENGTH_SHORT).show();
        } else {
            // Jumlah telur valid,
lanjutkan simpan data penjualan
            String idPenjualan =
pemasukanRef.push().getKey(); //
Generate unique key
            PenjualanData penjualan =
new PenjualanData(idPenjualan,
jumlahTelur, hargaSatuan, totalHarga,
tanggalMasukPenjualan, jenisTelur);

            // Simpan data penjualan

pemasukanRef.child(idPenjualan).setValue
(penjualan)

.addOnSuccessListener(new
OnSuccessListener<Void>() {
            @Override
            public void
onSuccess(Void aVoid) {

Toast.makeText(Tambah_penjualan.this,
"Data penjualan berhasil disimpan",
Toast.LENGTH_SHORT).show();

updateTotalTelur(jumlahTelur, jenisTelur);
// Update total telur di node total_telur

setResult(RESULT_OK); // Mengirimkan
kode RESULT_OK ke PenjualanActivity

```

```

Intent intent =
new Intent(Tambah_penjualan.this,
PenjualanActivity.class);

startActivity(intent);
        finish();
    }
})

.addOnFailureListener(new
OnFailureListener() {
        @Override
        public void
onFailure(@NonNull Exception e) {

Toast.makeText(Tambah_penjualan.this,
"Gagal menyimpan data: " +
e.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    });

}

}

@Override
public void
onCancelled(@NonNull DatabaseError
error) {
    // Handle error

Toast.makeText(Tambah_penjualan.this,
"Error: " + error.getMessage(),
Toast.LENGTH_SHORT).show();
    }
});
}

private void updateTotalTelur(int
jumlahTelurTerjual, String jenisTelur) {
    // Referensi ke node yang sesuai
berdasarkan jenis telur
    DatabaseReference telurRef =
FirebaseDatabase.getInstance().getReferen
ce()

```

```
.child("admin").child("total_telur").child(jenisTelur);
```

```
// Ambil total telur sekarang
```

```
telurRef.addListenerForSingleValueEvent(
    new ValueEventListener() {
```

```
        @Override
```

```
        public void
```

```
onDataChange(@NonNull DataSnapshot
    snapshot) {
```

```
    if (snapshot.exists()) {
```

```
        long totalTelurSekarang =
```

```
snapshot.getValue(Long.class);
```

```
        // Kurangi total telur dengan
```

```
jumlahTelurTerjual
```

```
        long totalTelurBaru =
```

```
totalTelurSekarang - jumlahTelurTerjual;
```

```
        // Update total telur di
```

```
Firebase
```

```
telurRef.setValue(totalTelurBaru)
```

```
.addOnSuccessListener(new
```

```
OnSuccessListener<Void>() {
```

```
    @Override
```

```
    public void
```

```
onSuccess(Void aVoid) {
```

```
        // Berhasil update
```

```
total telur
```

```
    }
```

```
});
```

```
.addOnFailureListener(new
```

```
OnFailureListener() {
```

```
    @Override
```

```
    public void
```

```
onFailure(@NonNull Exception e) {
```

```
        // Gagal update total
```

```
telur
```

```
    }
```

```
});
```

```
} else {
```

```
    // Node tidak ada, handle
```

dengan baik sesuai kebutuhan

```
Toast.makeText(Tambah_penjualan.this,
    "Node tidak ditemukan",
```

```
    Toast.LENGTH_SHORT).show();
```

```
    }
```

```
    }
```

```
        @Override
```

```
        public void
```

```
onCancelled(@NonNull DatabaseError
    error) {
```

```
        // Handle error
```

```
Toast.makeText(Tambah_penjualan.this,
    "Error: " + error.getMessage(),
```

```
    Toast.LENGTH_SHORT).show();
```

```
    }
```

```
});
```

```
}
```

```
public void onBackPressed(View view) {
```

```
    Intent intent = new
```

```
Intent(Tambah_penjualan.this,
```

```
PenjualanActivity.class);
```

```
    startActivity(intent);
```

```
    finish();
```

```
}
```

```
@SuppressWarnings("MissingSuperCall")
```

```
@Override
```

```
public void onBackPressed() {
```

```
    Intent intent = new
```

```
Intent(Tambah_penjualan.this,
```

```
PenjualanActivity.class);
```

```
intent.addFlags(Intent.FLAG_ACTIVITY_
    CLEAR_TOP |
```

```
Intent.FLAG_ACTIVITY_NEW_TASK);
```

```
    startActivity(intent);
```

```
    finish();
```

```
}
```

```
}
```

TELUR

DataKandang.java

```
package
com.aplikasi.apliaksiternaktelur.admin;

public class DataKandang {
    private String id;
    private String tanggal;
    private int jumlahAyam;
    private String keterangan;

    public DataKandang() {
        // Default constructor required for
        Firebase
    }

    public DataKandang(String id, String
    tanggal, int jumlahAyam, String
    keterangan) {
        this.id = id;
        this.tanggal = tanggal;
        this.jumlahAyam = jumlahAyam;
        this.keterangan = keterangan;
    }

    public String getId() {
        return id;
    }

    public String getTanggal() {
        return tanggal;
    }

    public int getJumlahAyam() {
        return jumlahAyam;
    }

    public String getKeterangan() {
        return keterangan;
    }
}
```

KandangActivity.java

```
package
com.aplikasi.apliaksiternaktelur.admin;

import android.os.Bundle;
import android.util.Log;
import android.widget.TextView;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;
import
androidx.recyclerview.widget.LinearLayoutManager;
import
androidx.recyclerview.widget.RecyclerView;

import com.aplikasi.apliaksiternaktelur.R;
import
com.google.firebase.database.DataSnapsho
t;
import
com.google.firebase.database.DatabaseErr
or;
import
com.google.firebase.database.DatabaseRef
erence;
import
com.google.firebase.database.FirebaseData
base;
import
com.google.firebase.database.ValueEventL
istener;

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

public class KandangActivity extends
AppCompatActivity {

    private TextView
    txtJumlahAyamSaatIni,txtJumlahPengeluar
    an;
    private RecyclerView
    recyclerViewKandang;
```

```

private KandangAdapter adapter;
private List<DataKandang>
dataKandangList;

private FirebaseDatabase database;
private DatabaseReference
adminDataKandangRef;
private DatabaseReference
userDataKandangRef;
private DatabaseReference
adminPengeluaranRef;
@Override
protected void onCreate(Bundle
savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_kandang
);

    txtJumlahPengeluaran =
findViewById(R.id.txtJumlahPengeluaran
);
    txtJumlahAyamSaatIni =
findViewById(R.id.txtJumlahAyamSaatIni
);
    recyclerViewKandang =
findViewById(R.id.recyclerViewKandang)
;

    database =
FirebaseDatabase.getInstance();
    adminDataKandangRef =
database.getReference("admin/data_kanda
ng");
    userDataKandangRef =
database.getReference("user/data_kandang
");

    adminPengeluaranRef =
database.getReference("admin/pengeluara
n");

    dataKandangList = new

```

```

ArrayList<>();
    adapter = new
KandangAdapter(dataKandangList);

recyclerViewKandang.setLayoutManager(
new LinearLayoutManager(this));

recyclerViewKandang.setAdapter(adapter)
;

    fetchDataKandang();
    fetchDataPengeluaran();
}

private void fetchDataKandang() {

adminDataKandangRef.addValueEventList
ener(new ValueEventListener() {
    @Override
    public void
onDataChange(@NonNull DataSnapshot
dataSnapshot) {

        processDataSnapshot(dataSnapshot);
    }

    @Override
    public void
onCancelled(@NonNull DatabaseError
databaseError) {
        Log.e("Firebase", "Gagal
membaca data kandang admin",
databaseError.toException());
    }
});

userDataKandangRef.addValueEventListe
ner(new ValueEventListener() {
    @Override
    public void
onDataChange(@NonNull DataSnapshot
dataSnapshot) {

        processDataSnapshot(dataSnapshot);
    }
});

```

```

    }

    @Override
    public void
    onCancelled(@NonNull DatabaseError
    databaseError) {
        Log.e("Firebase", "Gagal
        membaca data kandang user",
        databaseError.toException());
    }
    });
}

private void fetchDataPengeluaran() {

adminPengeluaranRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void
    onDataChange(@NonNull DataSnapshot
    dataSnapshot) {

processPengeluaranSnapshot(dataSnapshot
);

    }

    @Override
    public void
    onCancelled(@NonNull DatabaseError
    databaseError) {
        Log.e("Firebase", "Gagal
        membaca data pengeluaran admin",
        databaseError.toException());
    }
    });

adminPengeluaranRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void
    onDataChange(@NonNull DataSnapshot
    dataSnapshot) {

processPengeluaranSnapshot(dataSnapshot
);

```

```

    }

    @Override
    public void
    onCancelled(@NonNull DatabaseError
    databaseError) {
        Log.e("Firebase", "Gagal
        membaca data pengeluaran user",
        databaseError.toException());
    }
    });
}

private void
processPengeluaranSnapshot(DataSnapsho
t dataSnapshot) {
    int totalPengeluaran = 0;

    for (DataSnapshot snapshot :
    dataSnapshot.getChildren()) {
        Integer jumlahPengeluaran =
        snapshot.child("jumlah_pengeluaran").get
        Value(Integer.class);

        // Check if jumlahPengeluaran is
        not null
        if (jumlahPengeluaran != null) {
            totalPengeluaran +=
            jumlahPengeluaran;
        }
    }

    // Set total pengeluaran to TextView
    txtJumlahPengeluaran.setText("Total
    Pengeluaran: " + totalPengeluaran);
}

private void
processDataSnapshot(DataSnapshot
dataSnapshot) {
    dataKandangList.clear();
    int totalJumlahAyam = 0;

    for (DataSnapshot snapshot :

```

```

dataSnapshot.getChildren()) {
    String id =
snapshot.child("id").getValue(String.class)
;
    String tanggal =
snapshot.child("tanggal").getValue(String.
class);
    int jumlahAyam =
snapshot.child("jumlah_ayam").getValue(I
nteger.class);
    String keterangan =
snapshot.child("keterangan").getValue(Stri
ng.class);

    if (id != null && tanggal != null
&& keterangan != null) {
        DataKandang dataKandang =
new DataKandang(id, tanggal,
jumlahAyam, keterangan);

dataKandangList.add(dataKandang);
        totalJumlahAyam +=
jumlahAyam;
    }
}

txtJumlahAyamSaatIni.setText("Jumlah
Ayam Saat Ini: " + totalJumlahAyam);
    adapter.notifyDataSetChanged();
}
}

```

KandangAdapter.java

```

package
com.aplikasi.apliaksiternaktelur.admin;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import
androidx.recyclerview.widget.RecyclerView

```

```

w;

import com.aplikasi.apliaksiternaktelur.R;

import java.util.List;

public class KandangAdapter extends
RecyclerView.Adapter<KandangAdapter.V
iewHolder> {

    private List<DataKandang>
dataKandangList;

    public
KandangAdapter(List<DataKandang>
dataKandangList) {
        this.dataKandangList =
dataKandangList;
    }

    @NonNull
    @Override
    public ViewHolder
onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
        View view =
LayoutInflater.from(parent.getContext()).in
flate(R.layout.item_kandang, parent,
false);
        return new ViewHolder(view);
    }

    @Override
    public void
onBindViewHolder(@NonNull
ViewHolder holder, int position) {
        DataKandang dataKandang =
dataKandangList.get(position);

holder.textViewTanggal.setText("Tanggal:
" + dataKandang.getTanggal());

holder.textViewJumlahAyam.setText("Jum
lah Ayam: " +
dataKandang.getJumlahAyam());

```

```
holder.textViewKeterangan.setText("Keterangan: " + dataKandang.getKeterangan());
}
```

```
@Override
public int getItemCount() {
    return dataKandangList.size();
}
```

```
public static class ViewHolder extends
RecyclerView.ViewHolder {
    TextView textViewTanggal,
    textViewJumlahAyam,
    textViewKeterangan;
```

```
    public ViewHolder(@NonNull View
itemView) {
        super(itemView);
        textViewTanggal =
itemView.findViewById(R.id.textViewTan
ggal);
        textViewJumlahAyam =
itemView.findViewById(R.id.textViewJum
lahAyam);
        textViewKeterangan =
itemView.findViewById(R.id.textViewKete
rangan);
    }
}
}
```

setting.java

```
package
com.aplikasi.apliaksiternaktelur.admin;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AlertDialog;
import
androidx.appcompat.app.AppCompatActivity;

import android.content.DialogInterface;
import android.content.Intent;
```

```
import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;
import android.widget.Toast;
```

```
import
com.aplikasi.apliaksiternaktelur.MainActiv
ity;
import com.aplikasi.apliaksiternaktelur.R;
import
com.aplikasi.apliaksiternaktelur.admin.aya
m.AyamActivity;
import
com.aplikasi.apliaksiternaktelur.admin.aya
m.TambahAyam;
import
com.aplikasi.apliaksiternaktelur.admin.lap
oran.StatistikActivity;
import
com.aplikasi.apliaksiternaktelur.admin.lap
oran.coba;
import
com.google.android.gms.tasks.OnFailureL
istener;
import
com.google.android.gms.tasks.OnSuccess
Listener;
import
com.google.firebase.database.DatabaseRef
erence;
import
com.google.firebase.database.FirebaseData
base;
```

```
public class Setting extends
AppCompatActivity {
```

```
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_setting);
```

```

        ImageButton btnstatistik =
        findViewById(R.id.btnstatistik);
        ImageButton btnberanda =
        findViewById(R.id.btnberanda);
        ImageButton btnakun =
        findViewById(R.id.btnakun);

        btnstatistik.setBackgroundResource(R.dra
        wable.ic_chart);

        btnberanda.setBackgroundResource(R.dra
        wable.ic_home);

        btnakun.setBackgroundResource(R.dra
        wable.ic_settings_aktif);

    }
    public void gotoberanda(View view) {
        Intent intent = new Intent(Setting.this,
        MainActivity.class);
        startActivity(intent);
        finish();
    }

    public void gotopengaturan(View view)
    {
        Intent intent = new Intent(Setting.this,
        Setting.class);
        startActivity(intent);
        finish();
    }
    public void tesstat(View view) {
        Intent intent = new Intent(Setting.this,
        coba.class);
        startActivity(intent);
        finish();
    }
    public void gotostatistik(View view) {
        Intent intent = new Intent(Setting.this,
        StatistikActivity.class);
        startActivity(intent);

```

```

        finish();
    }
    public void resetdata(View view) {
        // Tampilkan dialog konfirmasi
        new AlertDialog.Builder(this)
            .setTitle("Reset Database")
            .setMessage("Apakah Anda
            yakin ingin mereset semua data? Tindakan
            ini tidak dapat dibatalkan.")

        .setPositiveButton(android.R.string.yes,
        new DialogInterface.OnClickListener() {
            public void
            onClick(DialogInterface dialog, int which)
            {
                // User mengklik tombol Ya
                resetDataFirebase();
            }
        })

        .setNegativeButton(android.R.string.no,
        null)

        .show();
    }

    private void resetDataFirebase() {
        DatabaseReference adminRef =
        FirebaseDatabase.getInstance().getReferen
        ce().child("admin");

        adminRef.removeValue()
            .addOnSuccessListener(new
            OnSuccessListener<Void>() {
                @Override
                public void onSuccess(Void
                aVoid) {
                    Toast.makeText(Setting.this,
                    "Database berhasil di-reset",
                    Toast.LENGTH_SHORT).show();
                }
            })
            .addOnFailureListener(new
            OnFailureListener() {

```

```

        @Override
        public void
onFailure(@NonNull Exception e) {
            Toast.makeText(Setting.this,
"Gagal mereset database: " +
e.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    });
}
}

```

splash.java

```

package
com.aplikasi.apliaksiternaktelur.admin;

// Buat file SplashActivity.java
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.Window;
import android.view.WindowManager;

import androidx.annotation.Nullable;
import
androidx.appcompat.app.AppCompatActivity;

import
com.aplikasi.apliaksiternaktelur.MainActivity;
import com.aplikasi.apliaksiternaktelur.R;

public class splash extends
AppCompatActivity {

    private static final long
SPLASH_DELAY = 2000; // 3 detik

    @Override
    protected void onCreate(@Nullable
Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

        // Menghilangkan title bar
requestWindowFeature(Window.FEATUR
E_NO_TITLE);

        // Mengatur activity menjadi full
screen

getWindow().setFlags(WindowManager.L
ayoutParams.FLAG_FULLSCREEN,

WindowManager.LayoutParams.FLAG_F
ULLSCREEN);

```

```

setContentView(R.layout.activity_splash);
        new Handler().postDelayed(new
Runnable() {
            @Override
            public void run() {
                // Arahkan ke MainActivity
setelah SPLASH_DELAY
                Intent intent = new
Intent(splash.this, MainActivity.class);
                startActivity(intent);
                finish(); // Menutup
SplashActivity agar tidak kembali saat
tombol back ditekan
            }
        }, SPLASH_DELAY);
    }
}

```

UI

MainActivity.java

```

package com.aplikasi.apliaksiternaktelur;

import android.os.Bundle;

import
com.google.android.material.bottomnaviga
tion.BottomNavigationView;

import

```

```

androidx.appcompat.app.ActionBar;
import
androidx.appcompat.app.AppCompatActivity;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import
androidx.navigation.ui.AppBarConfiguration;
import
androidx.navigation.ui.NavigationUI;

```

```

import
com.aplikasi.apliaksiternaktelur.databinding.ActivityMainBinding;
import
com.google.firebase.database.DatabaseReference;
import
com.google.firebase.database.FirebaseDatabase;

```

```

public class MainActivity extends AppCompatActivity {

    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ActionBar actionBar =
        getSupportActionBar();
        if (actionBar != null) {
            actionBar.hide();
        }
        binding =
        ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        BottomNavigationView navView =
        findViewById(R.id.nav_view);
        // Passing each menu ID as a set of
        Ids because each
        // menu should be considered as top

```

level destinations.

```

        AppBarConfiguration
        appBarConfiguration = new
        AppBarConfiguration.Builder(
            R.id.navigation_home,
            R.id.navigation_dashboard,
            R.id.navigation_notifications)
            .build();
        NavController navController =
        Navigation.findNavController(this,
        R.id.nav_host_fragment_activity_main);

```

```

        NavigationUI.setupActionBarWithNavController(this, navController,
        appBarConfiguration);

```

```

        NavigationUI.setupWithNavController(binding.navView, navController);

```

```

    }

```

```

}

```