*PS : Musik bisa berarti mediaPlayer*

**File song.java**

package id.ifundip.servicedemo;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class Song {  
 private final int icon;  
 private final String title;  
 private final int file;

Memanggil package dan mengimport library yang diperlukan

Membuat variable untuk menyimpan icon, title, dan file

public Song(int icon, String title, int file) {  
 this.icon = icon;  
 this.title = title;  
 this.file = file;  
}  
  
public int getIcon() {  
 return icon;  
}  
  
  
public String getTitle() {  
 return title;  
}  
  
public int getFile() {  
 return file;  
}

Fungsi song yang yang mengisi data pada class dengan icon, judul, dan file.

Fungsi-fungsi yang mendapatkan icon, title, dan file

public static List<Song> getSongs(){  
 List<Song> songs = new ArrayList<>();  
 songs.add(new Song(R.drawable.*ic\_song\_file*,"Fanfare",R.raw.*fanfare*));  
 songs.add(new Song(R.drawable.*ic\_song\_file*,"Harmoni",R.raw.*harmoni*));  
 songs.add(new Song(R.drawable.*ic\_song\_file*,"Piano 1 Stanza",R.raw.*piano\_1\_stanza*));  
 songs.add(new Song(R.drawable.*ic\_song\_file*,"Piano 3 Stanza",R.raw.*piano\_3\_stanza*));  
 songs.add(new Song(R.drawable.*ic\_song\_file*,"Piano & Paduan Suara 1 Stanza",R.raw.*piano\_paduan\_suara\_1\_stanza*));  
 songs.add(new Song(R.drawable.*ic\_song\_file*,"Piano & Paduan Suara 3 Stanza",R.raw.*piano\_paduan\_suara\_3\_stanza*));  
 songs.add(new Song(R.drawable.*ic\_song\_file*,"Simphoni 1 Stanza",R.raw.*simphoni\_1\_stanza*));  
 songs.add(new Song(R.drawable.*ic\_song\_file*,"Simphoni 3 Stanza",R.raw.*simphoni\_3\_stanza*));  
 songs.add(new Song(R.drawable.*ic\_song\_file*,"Simphoni & Vokal 1 Stanza",R.raw.*simphoni\_vokal\_1\_stanza*));  
 songs.add(new Song(R.drawable.*ic\_song\_file*,"Simphoni & Vokal 3 Stanza",R.raw.*simphoni\_vokal\_3\_stanza*));  
 songs.add(new Song(R.drawable.*ic\_song\_file*,"Unisono 1 Stanza",R.raw.*unisono\_1\_stanza*));  
 songs.add(new Song(R.drawable.*ic\_song\_file*,"Unisono 1 Stanza",R.raw.*unisono\_3\_stanza*));  
 return songs;

Membuat list yang berisi lagu-lagu yang akan ditampilkan yang disimpan pada arraylist.

**File SoundService.java**

package id.ifundip.servicedemo;  
  
import android.app.Service;  
import android.content.Intent;  
import android.media.MediaPlayer;  
import android.os.IBinder;  
  
import androidx.annotation.Nullable;

Memanggil package dan mengimport library yang diperlukan

Membuat variable untuk menyimpan icon, title, dan file

public class SoundService extends Service {  
 static MediaPlayer *player*;  
  
 @Override  
 public int onStartCommand(Intent intent, int flags, int startId) {  
 int file = intent.getIntExtra("song", 0);  
 *player* = MediaPlayer.*create*(this, file);  
 *player*.setOnPreparedListener(this::onPrepared);  
  
 return *START\_STICKY*;  
 }

Ketika musik diplay atau memulai service, akan dilakukan intent.

Variabel file meyimpan data yang diambil dari intent

Membuat mediaplayer.

setOnPreparedListener(this::onPrepared);????

Mengembalikan service.

private void onPrepared(MediaPlayer mediaPlayer) {  
 mediaPlayer.start();  
}  
  
@Override  
public void onDestroy() {  
 super.onDestroy();  
 stopPlayer();  
}

Memutarkan mediaplayer.

OnDestroy, menghentikan service dan musik.

public static void pausePlayer(){  
 if(*player* != null){  
 *player*.pause();  
 }  
}

Mempause musik

public static void resumePlayer(){  
 if(*player* != null){  
 *player*.start();  
 }  
}

Melanjutkann kembali musik yang dipause

private void stopPlayer() {  
 if (*player* != null) {  
 *player*.release();  
 *player* = null;  
 }  
}

Menghentikan musik yang dijalankan

@Nullable  
 @Override  
 public IBinder onBind(Intent intent) {  
 return null;  
 }  
}

Tidak memerlukan binding sehingga mengembalikan null karena tidak memerlukan koneksi yang lama.

**SongAdapter.java**

package id.ifundip.servicedemo;  
  
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 java.util.List;

Memanggil package dan mengimport library yang diperlukan

Membuat variable untuk menyimpan icon, title, dan file

public class SongAdapter extends RecyclerView.Adapter<SongAdapter.ViewHolder>{  
  
 private final List<Song> songs;  
 private OnItemClickListener listener;

Membuat variiable yang dimana list<song> menyimpan lagu yang disimpan dalam bentuk list dan OnClickListener yang menerima inputan click pada item tertentu.

public static class ViewHolder extends RecyclerView.ViewHolder{  
 public ImageView icon;  
 public TextView title;  
  
 public ViewHolder(@NonNull View itemView) {  
 super(itemView);  
  
 icon = itemView.findViewById(R.id.*iconSong*);  
 title = itemView.findViewById(R.id.*titleSong*);  
 }  
}

ViewHolder menjelaskan tampilan item dan metadata tentang tempatnya.

Menyimpan icon dan title pada image dan text view.

Pada viewholder, icon dan title mendapatkan icon dan title pada file xml.

public SongAdapter(List<Song> songs, OnItemClickListener listener) {  
 this.songs = songs;  
 this.listener = listener;  
}

Fungsi yang menerima inputan pada item yang dipilih dan disimpan pada class tersebut.

@NonNull  
@Override  
public SongAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 return new ViewHolder(LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*song\_row\_items*,parent,false));  
}

Membuat view baru, dengan return yang diberikan.

Return nya adalah berupa viewholder yang berisi layoutinflater dengan musik yang dimainkan.

@Override  
public void onBindViewHolder(@NonNull SongAdapter.ViewHolder holder, int position) {  
 Song song = songs.get(position);  
 holder.icon.setImageResource(song.getIcon());  
 holder.title.setText(song.getTitle());  
  
 holder.itemView.setOnClickListener(v -> {  
 if(listener != null){  
 listener.onItemClick(position);  
 }  
 });  
}

Membuat tampilan yang berisi posisi tampilan dan icon dan judul lagu yang ditampilkan

@Override  
public int getItemCount() {  
 return songs.size();  
}  
  
interface OnItemClickListener{  
 void onItemClick(int position);  
}

**MainActivity.java**

package id.ifundip.servicedemo;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.content.Intent;  
import android.media.MediaPlayer;  
import android.os.Bundle;  
import android.provider.Settings;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.util.List;

Memanggil package dan mengimport library yang diperlukan

Membuat variable untuk menyimpan icon, title, dan file

public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
 private boolean isPlaying = false;  
 private int currentSong = 0;  
 private List<Song> songs;  
  
 private TextView tvTitle;

Ketika menerima sebuah inputan click.

Membuat variabel.

@Override  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);

}

Membuat tampilan pada activity main

RecyclerView rvSongs = findViewById(R.id.*rvSongs*);  
  
Button btnPrevious = findViewById(R.id.*btnPrevious*);  
Button btnPlayStop = findViewById(R.id.*buttonPlayStop*);  
Button btnNext = findViewById(R.id.*btnNext*);

tvTitle = findViewById(R.id.*tvTitle*);

Mengambil elemen pada file xml sesuai id.

songs = Song.*getSongs*();

Mengambil musik yang dijalankan.

SongAdapter songAdapter = new SongAdapter(songs, index -> {  
 currentSong = index;  
 playMusic(index);  
});  
rvSongs.setAdapter(songAdapter);  
rvSongs.setLayoutManager(new LinearLayoutManager(this));  
  
btnPrevious.setOnClickListener(this);  
btnPlayStop.setOnClickListener(this);  
btnNext.setOnClickListener(this);

public void playMusic(int index) {  
 if (isPlaying) {  
 stopMusic();  
 }  
 isPlaying = true;  
 tvTitle.setText(songs.get(index).getTitle());  
 Intent intent = new Intent(this, SoundService.class);  
 intent.putExtra("song", songs.get(index).getFile());  
 startService(intent);  
}

Menjalankan musik.

Putextra= menambahkan data.

Dan menjalankan service berdasarkan intent.

public void stopMusic() {  
 isPlaying = false;  
 stopService(new Intent(this, SoundService.class));  
}

Menghentikan service media player.

private void playPauseMusic() {  
 if(isPlaying){  
// stopMusic();  
 SoundService.*pausePlayer*();  
 isPlaying = false;  
 }else{  
// playMusic(currentSong);  
 SoundService.*resumePlayer*();  
 isPlaying = true;  
 }  
 }

Fungsi untuk menghentikan sementara musik yang berjalan, dan melanjutkaan kembali dengan resume media player

private void previousMusic() {  
 if(currentSong > 0){  
 currentSong--;  
 playMusic(currentSong);  
 }else {  
 Toast.*makeText*(this, "This is first song", Toast.*LENGTH\_SHORT*).show();  
 }  
}

Membuat fungsi yang memutarkan musik sebelumnya

public void nextMusic() {  
 if(currentSong < songs.size() - 1){  
 currentSong++;  
 playMusic(currentSong);  
 }else {  
 Toast.*makeText*(this, "This is last song", Toast.*LENGTH\_SHORT*).show();  
 }  
}

@Override  
public void onClick(View view) {  
 if(view.getId() == R.id.*btnPrevious*){  
 previousMusic();  
 }else if(view.getId() == R.id.*buttonPlayStop*){  
 playPauseMusic();  
 }else if(view.getId() == R.id.*btnNext*){  
 nextMusic();  
 }  
}

Menjalan fungsi tergantung pada input button yang ditekan prev, pause, atau next.

**KAMUS**

[**RecyclerView**](https://developer.android.com/reference/androidx/recyclerview/widget/RecyclerView?hl=id) adalah [ViewGroup](https://developer.android.com/reference/android/view/ViewGroup?hl=id) yang berisi tampilan yang sesuai dengan data Anda. ViewGroup sendiri juga merupakan tampilan, jadi Anda menambahkan RecyclerView ke tata letak dengan cara yang sama seperti menambahkan elemen UI lainnya.