

LAMPIRAN

1. *Coding MainActivity*

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        this.requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.activity_main);

        BottomNavigationView navigation = (BottomNavigationView)
        findViewById(R.id.navigation);

        navigation.setOnNavigationItemSelectedListener(new ItemSelectedListener() {
            FragmentTransaction transaction =
            getSupportFragmentManager().beginTransaction();
            transaction.replace(R.id.content, Home.newInstance());
            transaction.commit();

        });
    }
}
```

2. *Source Code Streaming radio*

```
public class Home extends Fragment {

    private Intent serviceIntent;
    private Button btnPlay;
    private static boolean isStreaming = true;
    private ProgressDialog pdBuff = null;
    private boolean mBufferBroadcastIsRegistered;

    private Button btnStop;
    private View frag;

    private PagerAdapter mPagerAdapter;
    public static Home newInstance() {
        Home fragment = new Home();
        return fragment;
    }

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

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
    container,
```

```

        Bundle savedInstanceState) {
            frag = inflater.inflate(R.layout.fragment_home, container,
false);
            initialisePaging();

            final Animation animScale =
AnimationUtils.loadAnimation(getContext(), R.anim.sda);

            btnPlay = (Button) frag.findViewById(R.id.btnPlay);
            btnStop = (Button) frag.findViewById(R.id.btnStop);

            btnStop.setOnClickListener(new View.OnClickListener() {

                @Override
                public void onClick(View view) {
                    if (view == btnStop) {
                        Log.d("playStatus", "" + isStreaming);
                        view.startAnimation(animScale);
                        btnStop.setEnabled(false);
                        btnPlay.setEnabled(true);

            btnPlay.setBackgroundResource(R.drawable.ic_play);

            btnStop.setBackgroundResource(R.drawable.ic_stop_full);
                stopStreaming();
                isStreaming = true;
                Utils.setDataBooleanToSP(getActivity(),
Utils.IS_STREAM, false);

                    }

                }

            });
            btnPlay.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    if (view == btnPlay) {
                        Log.d("playStatus", "" + isStreaming);
                        view.startAnimation(animScale);
                        btnStop.setEnabled(true);
                        btnPlay.setEnabled(false);

            btnPlay.setBackgroundResource(R.drawable.ic_playfull);

            btnStop.setBackgroundResource(R.drawable.ic_stop);

                startStreaming();
                Utils.setDataBooleanToSP(getActivity(),
Utils.IS_STREAM, true);

                    }

                }

            });
        }
    }
}

```

3. Source Code StreamService

```

public class StreamService extends Service implements
    MediaPlayer.OnCompletionListener,
    MediaPlayer.OnPreparedListener,
    MediaPlayer.OnErrorListener,
    MediaPlayer.OnBufferingUpdateListener {

    public static final String URL_STREAM =
"http://78.129.224.21:36469";

    // notification
    private static final int NOTIFICATION_ID = 1;
    private PhoneStateListener phoneStateListener;
    private TelephonyManager telephonyManager;
    private boolean isPausedInCall = false;
    private NotificationCompat.Builder builder;
    private Intent bufferIntent;
    public static final String BROADCAST_BUFFER =
"com.bingen.radiobingen.broadcastbuffer";
    private MediaPlayer mediaPlayer = new MediaPlayer();

    @Override
    public void onCreate() {
        super.onCreate();
        Log.d("create", "service created");
        bufferIntent = new Intent(BROADCAST_BUFFER);
        mediaPlayer.setOnCompletionListener(this);
        mediaPlayer.setOnPreparedListener(this);
        mediaPlayer.setOnErrorListener(this);
        mediaPlayer.setOnBufferingUpdateListener(this);
        mediaPlayer.reset();
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId)
    {
        Log.d("play", "play streaming");
        telephonyManager = (TelephonyManager)
getSystemService(Context.TELEPHONY_SERVICE);
        phoneStateListener = new PhoneStateListener() {
            @Override
            public void onCallStateChanged(int state, String
incomingNumber) {
                switch (state) {
                    case TelephonyManager.CALL_STATE_OFFHOOK:
                    case TelephonyManager.CALL_STATE_RINGING:
                        if (mediaPlayer != null) {
                            pauseMedia();
                            isPausedInCall = true;
                        }
                }
                break;
            }
        };
    }
}

```

```

        phoneStateListener = new PhoneStateListener() {
            @Override
            public void onCallStateChanged(int state, String
incomingNumber) {
                switch (state) {
                    case TelephonyManager.CALL_STATE_OFFHOOK:
                    case TelephonyManager.CALL_STATE_RINGING:
                        if (mediaPlayer != null) {
                            pauseMedia();
                            isPausedInCall = true;
                        }
                        break;
                    case TelephonyManager.CALL_STATE_IDLE:
                        if (mediaPlayer != null) {
                            if (isPausedInCall) {
                                isPausedInCall = false;
                                playMedia();
                            }
                        }
                        break;
                }
            }
        };

        telephonyManager.listen(phoneStateListener,
PhoneStateListener.LISTEN_CALL_STATE);
        initNotification();
        mediaPlayer.reset();

        //play media
        if (!mediaPlayer.isPlaying()) {
            try {
                Log.d("streammm", "" + URL_STREAM);
                mediaPlayer.setDataSource(URL_STREAM);
                // sent to UI radio is buffer
                sendBufferingBroadcast();
                mediaPlayer.prepareAsync();
            } catch (IllegalArgumentException e) {
                Log.d("error", e.getMessage());
            } catch (IllegalStateException e) {
                Log.d("error", e.getMessage());
            } catch (IOException e) {
                Log.d("error", e.getMessage());
            }
        }
        return START_STICKY;
    }
}

```

4. Source Code DetailActivity

```

public class DetailActivity extends AppCompatActivity {

    TextView title_shows, ket_shows, harii, jamm;
    ImageView img;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.detail_shows);

        title_shows= (TextView) findViewById(R.id.title_shows);
        ket_shows= (TextView) findViewById(R.id.ket_shows);
        harii = (TextView) findViewById(R.id.hari);
        jamm = (TextView) findViewById(R.id.jam);
        img= (ImageView) findViewById(R.id.data_shows);

        Intent i = this.getIntent();
        String title = i.getExtras().getString("TITLE_KEY");
        String hari = i.getExtras().getString("HARI_KEY");
        String jam = i.getExtras().getString("JAM_KEY");
        String keterangan= i.getExtras().getString("KETERANGAN_KEY");
        String imga = i.getExtras().getString("IMG_KEY");

        title_shows.setText(title);
        harii.setText(hari);
        jamm.setText(jam);
        ket_shows.setText(keterangan);
        PicassoClient.downloadImage(this, imga, img);
    }
}

```

5. Source Code DataParse

```

public class DataParser extends AsyncTask<Void,Void,Boolean> {

    Context c;
    String jsonData;
    ListView lv;
    ProgressDialog pd;
    ArrayList<Shows> showss = new ArrayList<>();

    public DataParser(Context c, String jsonData, ListView lv) {
        this.c = c;
        this.jsonData = jsonData;
        this.lv = lv;
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }
}

```

```

        pd = new ProgressDialog(c);
        pd.setTitle("Parse");
        pd.setMessage("Parsing...Please Wait");
        pd.show();
    }

    @Override
    protected Boolean doInBackground(Void... parses) {
        return this.parseData();
    }

    @Override
    protected void onPostExecute(Boolean parsed) {
        super.onPostExecute(parsed);
        pd.dismiss();
        if(parsed){
            CustomAdapter adapter = new CustomAdapter(c, showss);
            lv.setAdapter(adapter);
        }else {
            Toast.makeText(c, "Tidak ada koneksi", Toast.LENGTH_SHORT).show();
        }
    }

    private Boolean parseData() {
        try {
            JSONArray ja = new JSONArray(jsonData);
            JSONObject jo;
            showss.clear();
            Shows shows;
            for(int i=0; i < ja.length(); i++){
                jo = ja.getJSONObject(i);
                int id = jo.getInt("Id_Shows");
                String title = jo.getString("Nama_Shows");
                String hari = jo.getString("Hari");
                String jam = jo.getString("Jam_Siaran");
                String keterangan = jo.getString("Deskripsi_Shows");
                String img = jo.getString("Img_Shows");
                shows = new Shows();
                shows.setId_shows(id);
                shows.setNama_shows(title);
                shows.setHari(hari);
                shows.setJam_siaran(jam);
                shows.setDeskripsi_shows(keterangan);
                shows.setImg(img);
                showss.add(shows);
            }
            return true;
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return false;
    }
}

```

6. Source Code CustomAdapter

```

public class CustomAdapter extends BaseAdapter {
    Context c;
    ArrayList<Shows> showss;

    public CustomAdapter(Context c, ArrayList<Shows> showss) {
        this.c = c;
        this.showss = showss;
    }

    @Override
    public int getCount() {
        return showss.size();
    }

    @Override
    public Object getItem(int position) {
        return showss.get(position);
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup
parent) {
        if (convertView == null) {
            convertView =
LayoutInflater.from(c).inflate(R.layout.model, parent, false);
        }

        TextView title_shows = (TextView)
convertView.findViewById(R.id.title_shows);
        ImageView img = (ImageView)
convertView.findViewById(R.id.data_shows);
        final Shows s = (Shows) this.getItem(position);
        title_shows.setText(s.getNama_shows());
        PicassoClient.downloadImage(c, s.getImg(), img);

        convertView.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View view) {

                openDetailActivity(s.getNama_shows(), s.getHari(), s.getJam_siaran(), s.
getDeskripsi_shows(), s.getImg());

            }
        });
        return convertView;
    }
}

```

7. Source Code Picasso Client

```

public class PicassoClient {

    public static void downloadImage(Context c, String imgUrl,
    ImageView img){
        if(imgUrl != null && imgUrl.length()>0){

Picasso.with(c).load(imgUrl).placeholder(R.drawable.placeholder).into
(img);
        }else {

Picasso.with(c).load(R.drawable.abc_action_bar_item_background_materi
al).into(img);

        }
    }
}

```

8. Source Code PHP

```

<?php
$host='localhost';
$username='*****';
$password='*****';
$db='id2490891_bingen';

$con=mysqli_connect($host,$username,$password,$db)or die('unable to connect');
if(mysqli_connect_error($con))
{
    echo"Failed to connect database".mysqli_connect_error();
}
$sql="SELECT * FROM `shows` ";
$result=mysqli_query($con,$sql);
if($result)
{
    while($row=mysqli_fetch_array($result))
    {
        $flag[]=$row;
    }
    print(json_encode($flag));
}
mysqli_close($con);
?>

```