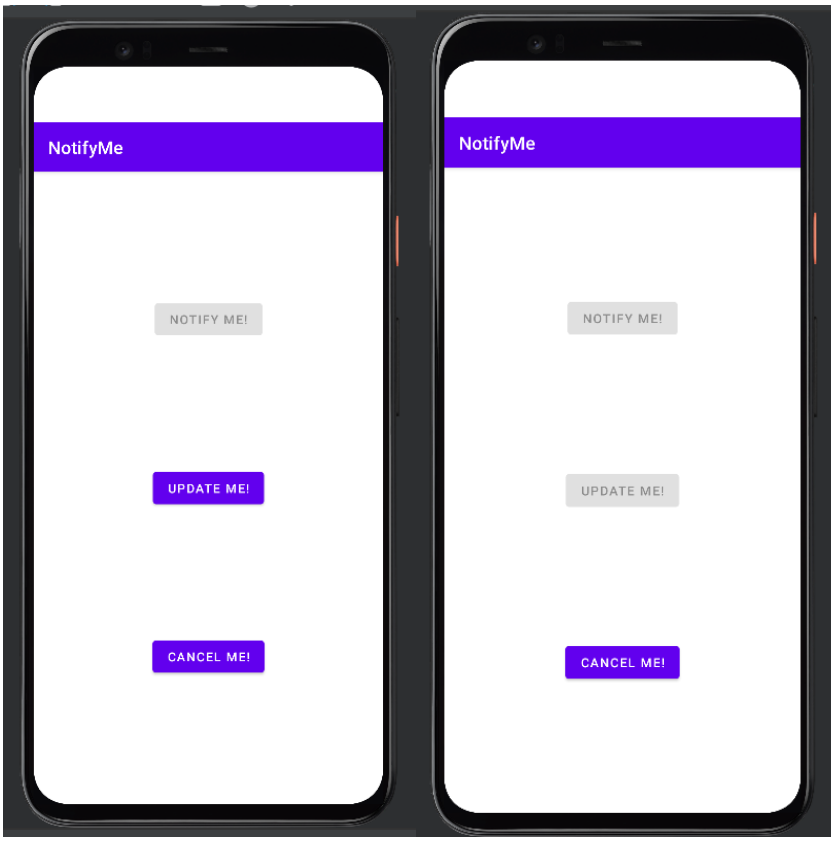
Mata Kuliah : Pemgraman Mobile – TI – S1

Pertemuan : Minggu 6

NIM : A11.2021.13509

Nama : Ainindzi Nur Meiza Pudjianto

1. **App NotifyMe**



- **MainActivity.java**

```
package com.example.notifyme;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
```

```
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    // Constants for the notification actions buttons.
    private static final String ACTION_UPDATE_NOTIFICATION =
"com.android.ryn.notifyme.ACTION_UPDATE_NOTIFICATION";
    // Notification channel ID.
    private static final String PRIMARY_CHANNEL_ID =
"primary_notification_channel";
    // Notification ID.
    private static final int NOTIFICATION_ID = 0;
    private Button button_notify;
    private Button button_cancel;
    private Button button_update;
    private NotificationManager mNotifyManager;
    private NotificationReceiver mReceiver = new NotificationReceiver();
    /**
     * Initializes the activity.
     *
     * @param savedInstanceState The current state data.
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
// Create the notification channel.
        createNotificationChannel();
// Register the broadcast receiver to receive the update actionfrom
// the notification.
        registerReceiver(mReceiver, new
IntentFilter(ACTION_UPDATE_NOTIFICATION));
// Add onClick handlers to all the buttons.
        button_notify = findViewById(R.id.notify);
        button_notify.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
// Send the notification
                sendNotification();
            }
        });
        button_update = (Button) findViewById(R.id.update);
        button_update.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
// Update the notification.
                updateNotification();
            }
        });
        button_cancel = (Button) findViewById(R.id.cancel);
        button_cancel.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
// Cancel the notification.
                cancelNotification();
```

```
            }
        });
// Reset the button states. Enable only Notify button and disable
// update and cancel buttons.
        setNotificationButtonState(true, false, false);
    }
    /**
     * Unregisters the receiver when the app is being destroyed.
     */
    @Override
    protected void onDestroy() {
        unregisterReceiver(mReceiver);
        super.onDestroy();
    }
    /**
     * Creates a Notification channel, for OREO and higher.
     */
    public void createNotificationChannel() {
// Create a notification manager object.
        mNotifyManager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
// Notification channels are only available in OREO and higher.
// So, add a check on SDK version.
        if (android.os.Build.VERSION.SDK_INT >=
android.os.Build.VERSION_CODES.O) {
// Create the NotificationChannel with all the parameters.
            NotificationChannel notificationChannel = new NotificationChannel
                    (PRIMARY_CHANNEL_ID,
                            getString(R.string.notification_channel_name),
                            NotificationManager.IMPORTANCE_HIGH);

            notificationChannel.enableLights(true);
            notificationChannel.setLightColor(Color.RED);
            notificationChannel.enableVibration(true);
            notificationChannel.setDescription
                    (getString(R.string.notification_channel_description));
            mNotifyManager.createNotificationChannel(notificationChannel);
        }
    }
    /**
     * OnClick method for the "Notify Me!" button.
     * Creates and delivers a simple notification.
     */
    public void sendNotification() {
// Sets up the pending intent to update the notification.
// Corresponds to a press of the Update Me! button.
        Intent updateIntent = new Intent(ACTION_UPDATE_NOTIFICATION);
        PendingIntent updatePendingIntent = PendingIntent.getBroadcast(this,
                        NOTIFICATION_ID, updateIntent,
                        PendingIntent.FLAG_ONE_SHOT);
// Build the notification with all of the parameters using helper
// method.
        NotificationCompat.Builder notifyBuilder = getNotificationBuilder();
// Add the action button using the pending intent.
        notifyBuilder.addAction(R.drawable.ic_update,
getString(R.string.update), updatePendingIntent);
// Deliver the notification.
```

```
        mNotifyManager.notify(NOTIFICATION_ID, notifyBuilder.build());
// Enable the update and cancel buttons but disables the "Notify
// Me!" button.
        setNotificationButtonState(false, true, true);
    }
    /**
     * Helper method that builds the notification.
     *
     * @return NotificationCompat.Builder: notification build with all the
     * parameters.
     */
    private NotificationCompat.Builder getNotificationBuilder() {
// Set up the pending intent that is delivered when the notification
// is clicked.
        Intent notificationIntent = new Intent(this, MainActivity.class);
        PendingIntent notificationPendingIntent = PendingIntent.getActivity
                (this, NOTIFICATION_ID, notificationIntent,
                        PendingIntent.FLAG_UPDATE_CURRENT);
// Build the notification with all of the parameters.
        NotificationCompat.Builder notifyBuilder = new NotificationCompat
                .Builder(this, PRIMARY_CHANNEL_ID)
                .setContentTitle(getString(R.string.notification_title))
                .setContentText(getString(R.string.notification_text))
                .setSmallIcon(R.drawable.ic_android)

.setAutoCancel(true).setContentIntent(notificationPendingIntent)
                .setPriority(NotificationCompat.PRIORITY_HIGH)
                .setDefaults(NotificationCompat.DEFAULT_ALL);
        return notifyBuilder;
    }
    /**
     * OnClick method for the "Update Me!" button. Updates the existing
     * notification to show a picture.
     */
    public void updateNotification() {
// Load the drawable resource into the a bitmap image.
        Bitmap androidImage = BitmapFactory
                .decodeResource(getResources(), R.drawable.mascot_1);
// Build the notification with all of the parameters using helper
// method.
        NotificationCompat.Builder notifyBuilder =
                getNotificationBuilder();
// Update the notification style to BigPictureStyle.
        notifyBuilder.setStyle(new NotificationCompat.BigPictureStyle()
                .bigPicture(androidImage)

.setBigContentTitle(getString(R.string.notification_updated)));
// Deliver the notification.
        mNotifyManager.notify(NOTIFICATION_ID, notifyBuilder.build());
        // Disable the update button, leaving only the cancel button enabled.
                setNotificationButtonState(false, false, true);
    }
    /**
     * OnClick method for the "Cancel Me!" button. Cancels the
     notification.
     */
    public void cancelNotification() {
```

```
// Cancel the notification.
        mNotifyManager.cancel(NOTIFICATION_ID);
// Reset the buttons.
        setNotificationButtonState(true, false, false);
    }
    /**
     * Helper method to enable/disable the buttons.
     *
     * @param isNotifyEnabled, boolean: true if notify button enabled
     * @param isUpdateEnabled, boolean: true if update button enabled
     * @param isCancelEnabled, boolean: true if cancel button enabled
     */
    void setNotificationButtonState(Boolean isNotifyEnabled, Boolean
            isUpdateEnabled, Boolean isCancelEnabled) {
        button_notify.setEnabled(isNotifyEnabled);
        button_update.setEnabled(isUpdateEnabled);
        button_cancel.setEnabled(isCancelEnabled);
    }
    /**
     * The broadcast receiver class for notifications.
     * Responds to the update notification pending intent action.
     */
    public class NotificationReceiver extends BroadcastReceiver {
        public NotificationReceiver() {
        }
        /**
         * Receives the incoming broadcasts and responds accordingly.
         *
         * @param context Context of the app when the broadcast is
        received.
         * @param intent The broadcast intent containing the action.
         */
        @Override
        public void onReceive(Context context, Intent intent) {
// Update the notification.
            updateNotification();
        }
    }
}
```

- **Activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/notify"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/notify_me"
        app:layout_constraintBottom_toTopOf="@+id/update"
```

```
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:id="@+id/update"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/update_me"
        app:layout_constraintBottom_toTopOf="@+id/cancel"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/notify" />
    <Button
        android:id="@+id/cancel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/cancel_me"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/update" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

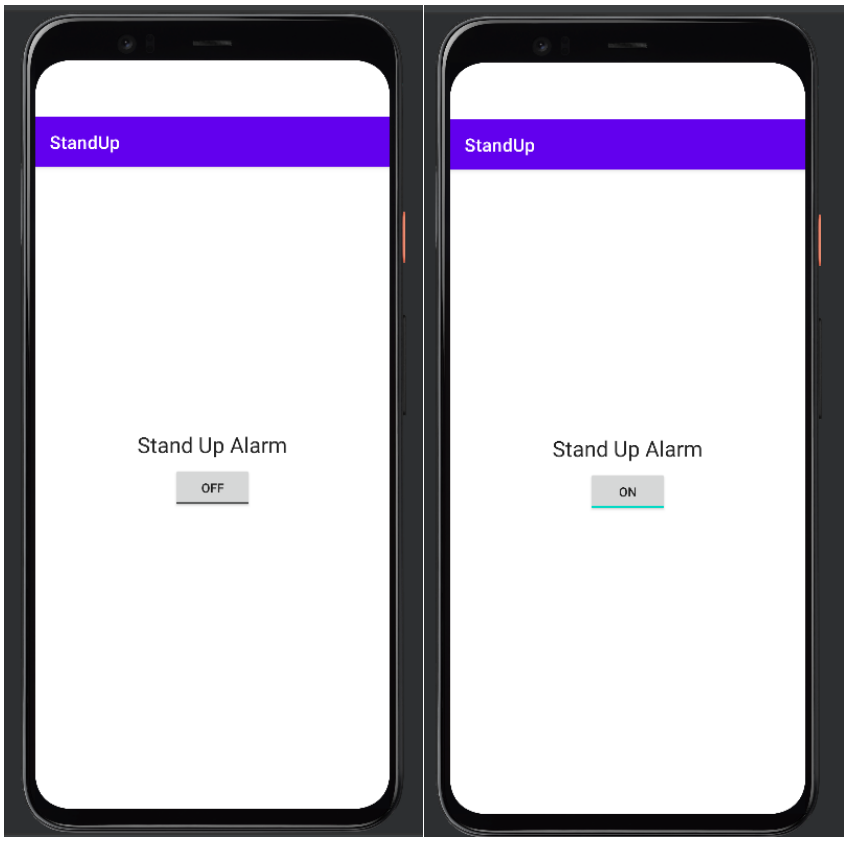
- **Colors.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
</resources>
```

- **Strings.xml**

```
<resources>
    <string name="app_name">NotifyMe</string>
    <string name="notification_title">You\'ve been notified!</string>
    <string name="notification_text">This is your notification text.</string>
    <string name="update">Update Notification</string>
    <string name="notify_me">Notify Me!</string>
    <string name="update_me">Update Me!</string>
    <string name="cancel_me">Cancel Me!</string>
    <string name="notification_updated">Notification Updated!</string>
    <string name="notification_channel_name">Mascot Notification</string>
    <string name="notification_channel_description">Notification from
Mascot</string>
</resources>
```

2. App StandUp



- **AlarmReceiver.java**

```
package com.example.standup;

import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;

import androidx.core.app.NotificationCompat;

public class AlarmReceiver extends BroadcastReceiver {
    private NotificationManager mNotificationManager;
    // Notification ID.
    private static final int NOTIFICATION_ID = 0;
    // Notification channel ID.
    private static final String PRIMARY_CHANNEL_ID =
            "primary_notification_channel";
    /**
     * Called when the BroadcastReceiver receives an Intent
     broadcast.
     *
     * @param context The Context in which the receiver is running.
     * @param intent The Intent being received.
```

```
     */
    @Override
    public void onReceive(Context context, Intent intent) {
        mNotificationManager = (NotificationManager)
                context.getSystemService(Context.NOTIFICATION_SERVICE);
// Deliver the notification.
        deliverNotification(context);
    }
    private void deliverNotification(Context context) {
// Create the content intent for the notification, whichlaunches
// this activity
        Intent contentIntent = new Intent(context,
                MainActivity.class);
        PendingIntent contentPendingIntent =
                PendingIntent.getActivity
                        (context, NOTIFICATION_ID, contentIntent,
                                PendingIntent.FLAG_UPDATE_CURRENT);
// Build the notification
        NotificationCompat.Builder builder = new NotificationCompat.Builder
                (context, PRIMARY_CHANNEL_ID)
                .setSmallIcon(R.drawable.ic_alarm)

.setContentTitle(context.getString(R.string.notification_title))

.setContentText(context.getString(R.string.notification_text))
                .setContentIntent(contentPendingIntent)
                .setPriority(NotificationCompat.PRIORITY_HIGH)
                .setAutoCancel(true)
                .setDefaults(NotificationCompat.DEFAULT_ALL);
// Deliver the notification
        mNotificationManager.notify(NOTIFICATION_ID,
                builder.build());
    }
}
```

- **Activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/alarmToggle"
        android:layout_centerHorizontal="true"
        android:layout_margin="8dp"
        android:text="@string/toggle_label"
```

```
        android:textAppearance="@style/TextAppearance.AppCompat.Headline"
        />
    <ToggleButton
        android:id="@+id/alarmToggle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true" />
</RelativeLayout>
```

- **Colors.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
</resources>
```

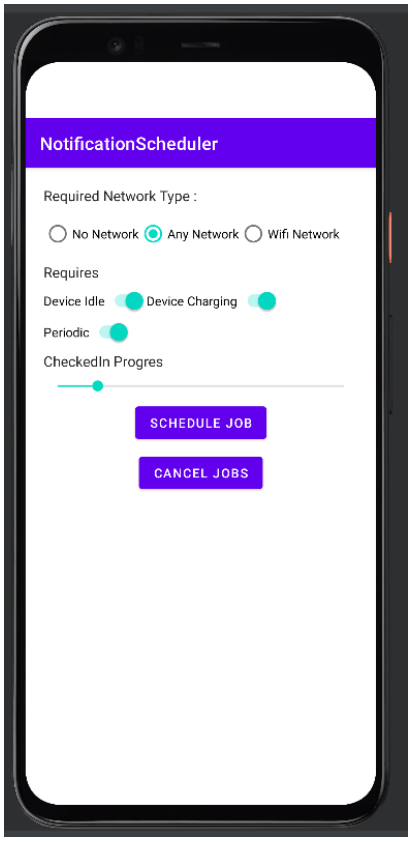
- **Dimens.xml**

```
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
</resources>
```

- **Strings.xml**

```
<resources>
    <string name="app_name">StandUp</string>
    <string name="alarm_on_toast">Stand Up Alarm On!</string>
    <string name="alarm_off_toast">Stand Up Alarm Off!</string>
    <string name="notification_title">Stand Up Alert</string>
    <string name="notification_text">You should stand up and walk around
now!</string>
    <string name="alarm_off">Alarm off</string>
    <string name="alarm_on">Alarm on</string>
    <string name="toggle_label">Stand Up Alarm</string>
</resources>
```

3. App Notification Schedular



- **MainActivity.java**

```
package com.example.notificationscheduler;

import androidx.appcompat.app.AppCompatActivity;

import android.app.job.JobInfo;
import android.app.job.JobScheduler;
import android.content.ComponentName;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.RadioGroup;
import android.widget.SeekBar;
import android.widget.Switch;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    private static final int JOB_ID = 0;
    private JobScheduler mScheduler;
    //Switches for setting job options
    private Switch mDeviceIdleSwitch;
    private Switch mDeviceChargingSwitch;
    private Switch mPeriodicSwitch;
    //Override deadline seekbar
    private SeekBar mSeekBar;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mDeviceIdleSwitch = (Switch) findViewById(R.id.idleSwitch);
        mDeviceChargingSwitch = (Switch) findViewById(R.id.chargingSwitch);
        mPeriodicSwitch = (Switch) findViewById(R.id.periodicSwitch);
        mSeekBar = (SeekBar) findViewById(R.id.seekBar);
        Button scheduleJobButton = (Button)
                findViewById(R.id.scheduleJobButton);
        Button cancelJobButton = (Button)
                findViewById(R.id.cancelJobsButton);
        final TextView label = (TextView) findViewById(R.id.seekBarLabel);
        final TextView seekBarProgress = (TextView)
                findViewById(R.id.seekBarProgress);
//Switch that toggles between periodic tasks and tasks with singledeadlines
        mPeriodicSwitch.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton compoundButton,
boolean isChecked) {
                if (isChecked){
                    label.setText(R.string.periodic_interval);
                } else {
                    label.setText(R.string.override_deadline);
                }
            }
        });
//Updates the TextView with the value from the seekbar
        mSeekBar.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onProgressChanged(SeekBar seekBar, int progress,
boolean userSet) {
                if (progress > 0){
                    String progressLabel = getString(R.string.seekbar_label,
progress);
                    seekBarProgress.setText(progressLabel);
                } else {
                    seekBarProgress.setText(R.string.not_set);
                }
            }
            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {
            }
            @Override
            public void onStopTrackingTouch(SeekBar seekBar) {
            }
        });
// OnClickListener that sets the job.
        scheduleJobButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                scheduleJob();
            }
        });
//OnClickListener that cancels all existing jobs.
```

```
        cancelJobButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                cancelJobs();
            }
        });
    }
    /**
     * onClick method that schedules the jobs based on the parameters set
     */
    private void scheduleJob() {
        mScheduler = (JobScheduler) getSystemService(JOB_SCHEDULER_SERVICE);
        RadioGroup networkOptions =
(RadioGroup)findViewById(R.id.NetworkOptions);

        int selectedNetworkID = networkOptions.getCheckedRadioButtonId();
        int selectedNetworkOption = JobInfo.NETWORK_TYPE_NONE;
        switch(selectedNetworkID){
            case R.id.noNetwork: selectedNetworkOption =
JobInfo.NETWORK_TYPE_NONE;
                break;
            case R.id.anyNetwork: selectedNetworkOption =
JobInfo.NETWORK_TYPE_ANY;
                break;
            case R.id.wifiNetwork: selectedNetworkOption =
JobInfo.NETWORK_TYPE_UNMETERED;
                break;
        }
        ComponentName serviceName = new ComponentName(getPackageName(),
                NotificationJobService.class.getName());
        JobInfo.Builder builder = new JobInfo.Builder(JOB_ID, serviceName)
                .setRequiredNetworkType(selectedNetworkOption)
                .setRequiresDeviceIdle(mDeviceIdleSwitch.isChecked())
                .setRequiresCharging(mDeviceChargingSwitch.isChecked());
        int seekBarInteger = mSeekBar.getProgress();
        boolean seekBarSet = seekBarInteger > 0;
//Set the job parameters based on the periodic switch.
        if (mPeriodicSwitch.isChecked()){
            if (seekBarSet){
                builder.setPeriodic(seekBarInteger * 1000);
            } else {
                Toast.makeText(MainActivity.this,
                        R.string.no_interval_toast,
                        Toast.LENGTH_SHORT).show();
            }
        } else {
            if (seekBarSet) {
                builder.setOverrideDeadline(seekBarInteger * 1000);
            }
        }
        boolean constraintSet = selectedNetworkOption !=
JobInfo.NETWORK_TYPE_NONE
                || mDeviceChargingSwitch.isChecked() ||
mDeviceIdleSwitch.isChecked()
                || seekBarSet;
        if(constraintSet) {
//Schedule the job and notify the user
```

```
            JobInfo myJobInfo = builder.build();
            mScheduler.schedule(myJobInfo);
            Toast.makeText(this, R.string.job_scheduled,
                    Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(this, R.string.no_constraint_toast,
                    Toast.LENGTH_SHORT).show();
        }
    }
    private void cancelJobs() {
        if (mScheduler != null){
            mScheduler.cancelAll();
            mScheduler = null;
            Toast.makeText(this, R.string.jobs_canceled,
                    Toast.LENGTH_SHORT).show();
        }
    }
}
```

- **NotificationJobService.java**

```
package com.example.notificationscheduler;

import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.job.JobParameters;
import android.app.job.JobService;
import android.content.Intent;

import androidx.core.app.NotificationCompat;

public class NotificationJobService extends JobService {
    @Override
    public boolean onStartJob(JobParameters jobParameters) {
//Set up the notification content intent to launch the app whenclicked
        PendingIntent contentPendingIntent = PendingIntent.getActivity
                (this, 0, new Intent(this, MainActivity.class),
                        PendingIntent.FLAG_UPDATE_CURRENT);
        NotificationManager manager = (NotificationManager)
                getSystemService(NOTIFICATION_SERVICE);
        NotificationCompat.Builder builder = new
                NotificationCompat.Builder(this)
                .setContentTitle(getString(R.string.job_service))
                .setContentText(getString(R.string.job_running))
                .setContentIntent(contentPendingIntent)
                .setSmallIcon(R.drawable.img)
                .setPriority(NotificationCompat.PRIORITY_HIGH)
                .setDefaults(NotificationCompat.DEFAULT_ALL)
                .setAutoCancel(true);
        manager.notify(0, builder.build());
        return false;
    }
    @Override
    public boolean onStopJob(JobParameters jobParameters) {
        return true;
```

```
    }
}
```

- **Activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/required_network_type"
        android:textAppearance="@style/TextAppearance.AppCompat.Subhead"
        android:layout_margin="4dp"/>
    <RadioGroup
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:id="@+id/NetworkOptions"
        android:layout_margin="4dp">
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/no_network"
            android:checked="true"
            android:id="@+id/noNetwork"/>
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/any_network"
            android:id="@+id/anyNetwork"/>
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/wifi_network"
            android:id="@+id/wifiNetwork"/>
    </RadioGroup>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/requires"
        android:textAppearance="@style/TextAppearance.AppCompat.Subhead"
        android:layout_margin="4dp"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_margin="4dp">
```

```
        <Switch
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/device_idle"
            android:id="@+id/idleSwitch"/>

        <Switch
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/device_charging"
            android:id="@+id/chargingSwitch"/>
    </LinearLayout>
    <Switch
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/periodic"
        android:id="@+id/periodicSwitch"
        android:layout_margin="4dp"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_margin="4dp">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/override_deadline"
            android:id="@+id/seekBarLabel"

android:textAppearance="@style/TextAppearance.AppCompat.Subhead"/>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/not_set"
            android:id="@+id/seekBarProgress"

android:textAppearance="@style/TextAppearance.AppCompat.Subhead"/>
    </LinearLayout>
    <SeekBar
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/seekBar"
        android:layout_margin="4dp"/>
    <Button
        android:id="@+id/scheduleJobButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="4dp"
        android:text="@string/schedule_job"
        android:layout_gravity="center_horizontal"/>
    <Button
        android:id="@+id/cancelJobsButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="4dp"
        android:text="@string/cancel_jobs"
```

```
        android:layout_gravity="center_horizontal"/>
</LinearLayout>
```

- **Colors.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
</resources>
```

- **Dimens.xml**

```
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
</resources>
```

- **Strings.xml**

```
<resources>
    <string name="app_name">NotificationScheduler</string>
    <string name="required_network_type">Required Network Type :</string>
    <string name="no_network">No Network</string>
    <string name="any_network">Any Network</string>
    <string name="wifi_network">Wifi Network</string>
    <string name="requires">Requires</string>
    <string name="device_idle">Device Idle</string>
    <string name="device_charging">Device Charging</string>
    <string name="periodic">Periodic</string>
    <string name="override_deadline">Override Deadline</string>
    <string name="not_set">Not Set</string>
    <string name="schedule_job">SCHEDULE JOB</string>
    <string name="cancel_jobs">CANCEL JOBS</string>
    <string name="periodic_interval">Checked</string>
    <string name="seekbar_label">In Progres</string>
    <string name="no_interval_toast">interval</string>
    <string name="job_scheduled">job</string>
    <string name="no_constraint_toast">constrait</string>
    <string name="jobs_canceled">canceled</string>
    <string name="job_service">in service</string>
    <string name="job_running">job is running</string>
</resources>
```

4. Aplikasi Nofitikasi dan Alarm untuk Jadwal Kuliah

- **MainActivity.java**

```
package com.example.pert6mei;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import android.app.AlarmManager;
import android.app.PendingIntent;
import android.app.TimePickerDialog;
import android.content.Context;
import android.content.Intent;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.TimePicker;
import java.util.Calendar;
public class MainActivity extends AppCompatActivity {
    Button buttonSetDialog;
    TextView textAlarm;
    TimePickerDialog timePickerDialog;
    final static int RQS = 1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textAlarm = (TextView)findViewById(R.id.alarm);
        buttonSetDialog = (Button)
```

```
                findViewById(R.id.btnStart);
        buttonSetDialog.setOnClickListener(new
                                                    View.OnClickListener() {
                                                        @Override
                                                        public void
onClick(View v) {

textAlarm.setText("");
                                                            Calendar cal =
Calendar.getInstance();
                                                            timePickerDialog =
new

TimePickerDialog(MainActivity.this,

onTimeSetListener,

cal.get(Calendar.HOUR_OF_DAY),

cal.get(Calendar.MINUTE),true);

timePickerDialog.setTitle("Set Alarm Time");

timePickerDialog.show();
                                                        }
                                                    });
    }
    TimePickerDialog.OnTimeSetListener onTimeSetListener =
            new
                    TimePickerDialog.OnTimeSetListener() {
                        @Override
                        public void onTimeSet(TimePicker view, int
                                hourOfDay, int minute) {
                            Calendar calNow =
                                    Calendar.getInstance();
                            Calendar calSet = (Calendar)
                                    calNow.clone();

                            calSet.set(Calendar.HOUR_OF_DAY,hourOfDay);
                            calSet.set(Calendar.MINUTE,minute);
                            calSet.set(Calendar.SECOND, 0);
                            calSet.set(Calendar.MILLISECOND,0);
                            if(calSet.compareTo(calNow) <= 0){
                                calSet.add(Calendar.DATE,1);
                                Log.i("hasil","=< 0");
                            }else if (calSet.compareTo(calNow) > 0){
                                Log.i("hasil","> 0");
                            }else {
                                Log.i("hasil"," else ");
                            }
                            textAlarm.setText("***\n" + "Alarm Set On " +
calSet.getTime() + "\n***");
                            Intent i = new Intent(getBaseContext(),

com.example.pert6mei.AlarmReceiver.class);
                            PendingIntent pi =
```

```
PendingIntent.getBroadcast(getBaseContext(),RQS,i,0);
                            AlarmManager almMgr = (AlarmManager)

                                    getSystemService(Context.ALARM_SERVICE);
                            long repeatInterval =
                                    almMgr.ELAPSED_REALTIME_WAKEUP;


almMgr.setInexactRepeating(AlarmManager.RTC_WAKEUP,calSet.getTimeInMillis(),r
epeatInterval,pi);
                        }
                    };
}
```

- **AlarmReceiver.java**

```
package com.example.pert6mei;
import static androidx.core.content.ContextCompat.getSystemService;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.MediaPlayer;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Build;
import android.widget.Toast;
import androidx.core.app.NotificationCompat;
public class AlarmReceiver extends BroadcastReceiver {
    MediaPlayer mp;
    private static final int NOTIFICATION_ID = 0;
    public static final String NOTIFICATION_CHANNEL_ID = "4655";
    @Override
    public void onReceive(Context context, Intent intent) {
        // TODO: This method is called when the BroadcastReceiver is
receiving
        // an Intent broadcast.
// throw new UnsupportedOperationException("Not yet implemented");
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            CharSequence name = NOTIFICATION_CHANNEL_ID;
            String description = context.getString(R.string.content);
            int importance = NotificationManager.IMPORTANCE_DEFAULT;
            NotificationChannel mChannel = new
NotificationChannel(NOTIFICATION_CHANNEL_ID, name, importance);

            mChannel.setDescription(description);
            // Register the channel with the system; you can't change the
importance
            // or other notification behaviors after this
            NotificationManager notificationManager =
context.getSystemService(NotificationManager.class);
            notificationManager.createNotificationChannel(mChannel);
```

```
        }

        //NotificationChannel mChannel = new
NotificationChannel(NOTIFICATION_CHANNEL_ID,
"notification",NotificationManager.IMPORTANCE_LOW);
        // mChannel.enableLights(true);

        Toast.makeText(context, "Alarm Aktif", Toast.LENGTH_SHORT).show();
        System.out.println("Jalan tidak!!!!");
        Uri notif =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
        Ringtone r = RingtoneManager.getRingtone(context, notif);
        r.play();
        NotificationManager notifMgr = (NotificationManager)
                context.getSystemService(Context.NOTIFICATION_SERVICE);
        Intent i = new Intent(context, MainActivity.class);
        PendingIntent pi =
                PendingIntent.getActivity(context, NOTIFICATION_ID, i,
PendingIntent.FLAG_UPDATE_CURRENT);
        NotificationCompat.Builder build = new
NotificationCompat.Builder(context,NOTIFICATION_CHANNEL_ID)
                .setSmallIcon(R.drawable.ic_alarm)
                .setVibrate(new long[]{100, 200, 300, 400, 500, 400, 300,
200, 400})
                .setContentTitle(context.getString(R.string.head))
                .setContentText(context.getString(R.string.content))
                .setContentIntent(pi)
                .setAutoCancel(false)
                .setPriority(NotificationCompat.PRIORITY_HIGH)
                .setDefaults(NotificationCompat.DEFAULT_ALL);
        notifMgr.notify(NOTIFICATION_ID, build.build());
        Toast.makeText(context, context.getString(R.string.content),
                Toast.LENGTH_LONG).show();
    }
}
```

- **Activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/tv1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Alarm Kuliah"

        android:textAppearance="?android:attr/textAppearanceLarge"/>
    <TextView
```

```
        android:id="@+id/tv2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Set Alarm 30 minute before start"

        android:textAppearance="?android:attr/textAppearanceSmall"/>
    <AnalogClock
        android:id="@+id/analog"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:drawableLeft="@android:drawable/ic_lock_idle_alarm"
        android:text=" Set Alarm"
        android:id="@+id/btnStart"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:id="@+id/alarm"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:text="Set Alarm again"
        android:id="@+id/tv3" />
</LinearLayout>
```

- **Colors.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
</resources>
```

- **Strings.xml**

```
<resources>
    <string name="app_name">MyAlarm</string>
    <string name="head">Alarm Kuliah</string>
    <string name="content">Ayoo Kuliah</string>
</resources>
```