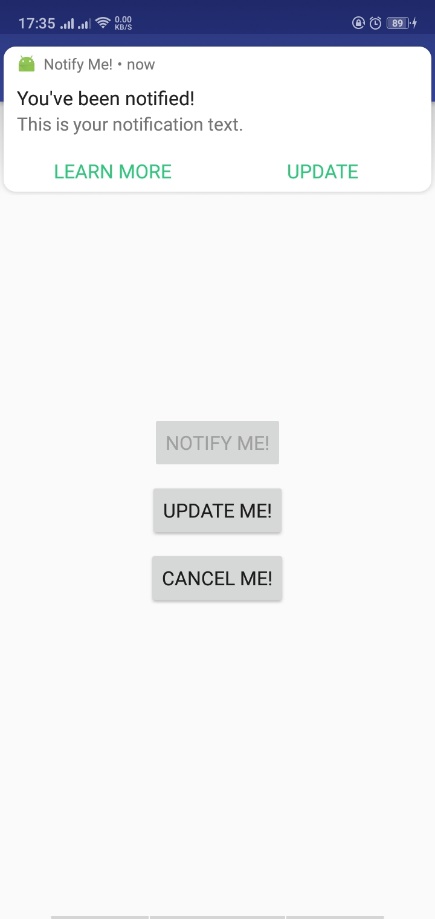
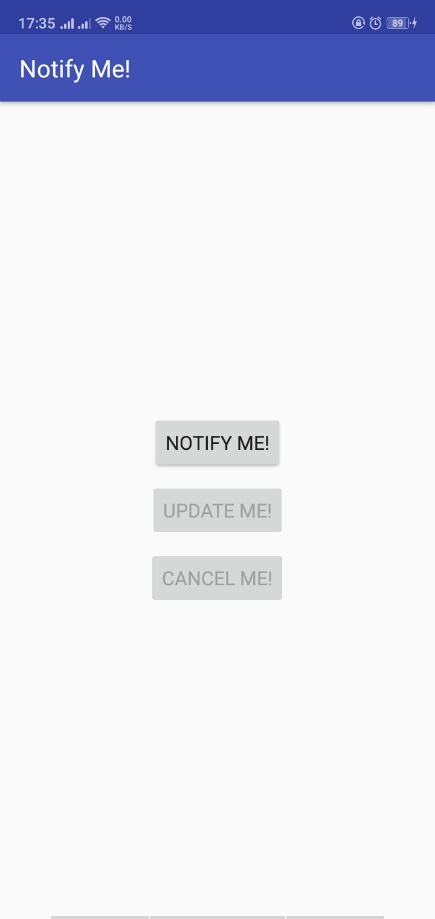
**Nama : Bayu Prasetya Adji Sugiyarto**

**NIM : A11.2019.11688**

**Pertemuan : 6**

**Kelompok : A11.4304**

**NotifyMe**

****

**String.xml**

<**resources**>  
 <**string name="app\_name"**>Notify Me!</**string**>  
 <**string name="notification\_title"**>You\'ve been notified!</**string**>  
 <**string name="notification\_text"**>This is your notification text.</**string**>  
 <**string name="learn\_more"**>Learn More</**string**>  
 <**string name="update"**>Update</**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**>  
</**resources**>

**Mainactivity.java**

**package** com.example.android.notifyme;  
  
**import** android.app.Notification;  
**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.net.Uri;  
**import** android.os.Bundle;  
**import** android.support.v4.app.NotificationCompat;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.view.View;  
**import** android.widget.Button;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
  
 **private** NotificationManager **mNotifyManager**;  
 **private static final int *NOTIFICATION\_ID*** = 0;  
 **private static final** String ***NOTIFICATION\_GUIDE\_URL*** =  
 **"https://developer.android.com/design/patterns/notifications.html"**;  
 **private static final** String ***ACTION\_UPDATE\_NOTIFICATION*** =  
 **"com.example.android.notifyme.ACTION\_UPDATE\_NOTIFICATION"**;  
 **private static final** String ***ACTION\_CANCEL\_NOTIFICATION*** =  
 **"com.example.android.notifyme.ACTION\_CANCEL\_NOTIFICATION"**;  
  
 **private** Button **mNotifyButton**;  
 **private** Button **mUpdateButton**;  
 **private** Button **mCancelButton**;  
  
 **private** NotificationReceiver **mReceiver** = **new** NotificationReceiver();  
   
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
  
 **mNotifyManager** = (NotificationManager) getSystemService(***NOTIFICATION\_SERVICE***);  
  
 **mNotifyButton** = (Button) findViewById(R.id.***notify***);  
 **mUpdateButton** = (Button) findViewById(R.id.***update***);  
 **mCancelButton** = (Button) findViewById(R.id.***cancel***);  
  
 **mNotifyButton**.setEnabled(**true**);  
 **mUpdateButton**.setEnabled(**false**);  
 **mCancelButton**.setEnabled(**true**);  
  
 *//Initialize and register the notification receiver* IntentFilter intentFilter = **new** IntentFilter();  
 intentFilter.addAction(***ACTION\_UPDATE\_NOTIFICATION***);  
 intentFilter.addAction(***ACTION\_CANCEL\_NOTIFICATION***);  
 registerReceiver(**mReceiver**, intentFilter);  
  
  
 *//Set OnClick methods.* **mNotifyButton**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 sendNotification();  
 }  
 });  
  
 **mUpdateButton**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 updateNotification();  
 }  
 });  
  
 **mCancelButton**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 cancelNotification();  
 }  
 });  
  
 }  
  
 @Override  
 **protected void** onDestroy() {  
 unregisterReceiver(**mReceiver**);  
 **super**.onDestroy();  
 }  
  
 **public void** sendNotification() {  
  
 *//Sets 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***);  
  
  
 *// Sets up the pending intent to cancel the notification,  
 // delivered when the user dismisses the notification* Intent cancelIntent = **new** Intent(***ACTION\_CANCEL\_NOTIFICATION***);  
 PendingIntent cancelPendingIntent = PendingIntent.*getBroadcast* (**this**, ***NOTIFICATION\_ID***, cancelIntent, PendingIntent.***FLAG\_ONE\_SHOT***);  
  
 *//Sets up the pending intent associated with the Learn More notification action,  
 //uses an implicit intent to go to the web.* Intent learnMoreIntent = **new** Intent(Intent.***ACTION\_VIEW***, Uri.*parse*(***NOTIFICATION\_GUIDE\_URL***));  
 PendingIntent learnMorePendingIntent = PendingIntent.*getActivity* (**this**, ***NOTIFICATION\_ID***, learnMoreIntent, PendingIntent.***FLAG\_ONE\_SHOT***);  
  
 *//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***);  
  
 *//Builds the notification with all of the parameters* NotificationCompat.Builder notifyBuilder = **new** NotificationCompat.Builder(**this**)  
 .setContentTitle(getString(R.string.***notification\_title***))  
 .setContentText(getString(R.string.***notification\_text***))  
 .setSmallIcon(R.drawable.***ic\_android***)  
 .setContentIntent(notificationPendingIntent)  
 .setPriority(NotificationCompat.***PRIORITY\_HIGH***)  
 .setDefaults(NotificationCompat.***DEFAULT\_ALL***)  
 .addAction(R.drawable.***ic\_learn\_more***, getString(R.string.***learn\_more***),  
 learnMorePendingIntent)  
 .addAction(R.drawable.***ic\_update***, getString(R.string.***update***), updatePendingIntent)  
 .setDeleteIntent(cancelPendingIntent);  
  
 *//Delivers the notification* **mNotifyManager**.notify(***NOTIFICATION\_ID***, notifyBuilder.build());  
  
 *//Enables the update and cancel buttons but disables the "Notify Me!" button* **mNotifyButton**.setEnabled(**false**);  
 **mUpdateButton**.setEnabled(**true**);  
 **mCancelButton**.setEnabled(**true**);  
  
 }  
  
 */\*\*  
 \* OnClick method for the "Update Me!" button. Updates the existing notification to show a  
 \* picture.  
 \*/* **private void** updateNotification() {  
  
 *//Load the drawable resource into the a bitmap image* Bitmap androidImage = BitmapFactory.*decodeResource*(getResources(),R.drawable.***mascot\_1***);  
  
 *//Sets 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***);  
  
  
 *// Sets up the pending intent to cancel the notification,  
 // delivered when the user dismisses the notification* Intent cancelIntent = **new** Intent(***ACTION\_CANCEL\_NOTIFICATION***);  
 PendingIntent cancelPendingIntent = PendingIntent.*getBroadcast* (**this**, ***NOTIFICATION\_ID***, cancelIntent, PendingIntent.***FLAG\_ONE\_SHOT***);  
  
 *//Sets up the pending intent associated with the Learn More notification action,  
 //uses an implicit intent to go to the web.* Intent learnMoreIntent = **new** Intent(Intent.***ACTION\_VIEW***, Uri.*parse*(***NOTIFICATION\_GUIDE\_URL***));  
 PendingIntent learnMorePendingIntent = PendingIntent.*getActivity* (**this**, ***NOTIFICATION\_ID***, learnMoreIntent, PendingIntent.***FLAG\_ONE\_SHOT***);  
  
 *//Build the updated notification* NotificationCompat.Builder notifyBuilder = **new** NotificationCompat.Builder(**this**)  
 .setContentTitle(getString(R.string.***notification\_title***))  
 .setContentText(getString(R.string.***notification\_text***))  
 .setSmallIcon(R.drawable.***ic\_android***)  
 .setContentIntent(notificationPendingIntent)  
 .setPriority(NotificationCompat.***PRIORITY\_DEFAULT***)  
 .setDefaults(NotificationCompat.***DEFAULT\_ALL***)  
 .setDeleteIntent(cancelPendingIntent)  
 .addAction(R.drawable.***ic\_learn\_more***, getString(R.string.***learn\_more***),  
 learnMorePendingIntent)  
 .setStyle(**new** NotificationCompat.BigPictureStyle()  
 .bigPicture(androidImage)  
 .setBigContentTitle(getString(R.string.***notification\_updated***)));  
  
  
 *//Disable the update button, leaving only the option to cancel* **mNotifyButton**.setEnabled(**false**);  
 **mUpdateButton**.setEnabled(**false**);  
 **mCancelButton**.setEnabled(**true**);  
  
 *//Deliver the notification* Notification myNotification = notifyBuilder.build();  
 **mNotifyManager**.notify(***NOTIFICATION\_ID***, myNotification);  
  
 }  
  
 */\*\*  
 \* OnClick method for the "Cancel Me!" button. Cancels the notification  
 \*/* **private void** cancelNotification() {  
 *//Cancel the notification* **mNotifyManager**.cancel(***NOTIFICATION\_ID***);  
  
 *//Resets the buttons* **mNotifyButton**.setEnabled(**true**);  
 **mUpdateButton**.setEnabled(**false**);  
 **mCancelButton**.setEnabled(**false**);  
 }  
  
  
 */\*\*  
 \* The broadcast receiver class for notifications. Responds to the update notification and  
 \* cancel notification pending intents actions.  
 \*/* **private class** NotificationReceiver **extends** BroadcastReceiver {  
  
 */\*\*  
 \* Gets the action from the incoming broadcast intent 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) {  
 String action = intent.getAction();  
 **switch** (action){  
 **case *ACTION\_CANCEL\_NOTIFICATION***:  
 cancelNotification();  
 **break**;  
 **case *ACTION\_UPDATE\_NOTIFICATION***:  
 updateNotification();  
 **break**;  
 }  
 }  
 }  
}

**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:gravity="center"  
 android:orientation="vertical"  
 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="com.example.android.notifyme.MainActivity"**>  
  
 <**Button  
 android:id="@+id/notify"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="4dp"  
 android:text="@string/notify\_me"** />  
  
 <**Button  
 android:id="@+id/update"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="4dp"  
 android:text="@string/update\_me"** />  
  
 <**Button  
 android:id="@+id/cancel"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="4dp"  
 android:text="@string/cancel\_me"** />  
</**LinearLayout**>

**StandUp**

****

**String.xml**

<**resources**>  
 <**string name="app\_name"**>Stand Up!</**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="toggle\_label"**>Stand Up Alarm</**string**>  
</**resources**>

**MainActivity.java**

**package** com.example.android.standup;  
  
**import** android.app.AlarmManager;  
**import** android.app.NotificationManager;  
**import** android.app.PendingIntent;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.os.SystemClock;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.widget.CompoundButton;  
**import** android.widget.Toast;  
**import** android.widget.ToggleButton;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 **private** NotificationManager **mNotificationManager**;  
  
 **private static final int *NOTIFICATION\_ID*** = 0;  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **mNotificationManager** = (NotificationManager) getSystemService(***NOTIFICATION\_SERVICE***);  
 **final** AlarmManager alarmManager = (AlarmManager) getSystemService(***ALARM\_SERVICE***);  
  
 ToggleButton alarmToggle = (ToggleButton) findViewById(R.id.***alarmToggle***);  
  
 *//Set up the Notification Broadcast Intent* Intent notifyIntent = **new** Intent(**this**, AlarmReceiver.**class**);  
  
 *//Check if the Alarm is already set, and check the toggle accordingly* **boolean** alarmUp = (PendingIntent.*getBroadcast*(**this**, 0, notifyIntent,  
 PendingIntent.***FLAG\_NO\_CREATE***) != **null**);  
  
 alarmToggle.setChecked(alarmUp);  
  
 *//Set up the PendingIntent for the AlarmManager* **final** PendingIntent notifyPendingIntent = PendingIntent.*getBroadcast* (**this**, ***NOTIFICATION\_ID***, notifyIntent, PendingIntent.***FLAG\_UPDATE\_CURRENT***);  
  
  
  
 alarmToggle.setOnCheckedChangeListener(**new** CompoundButton.OnCheckedChangeListener() {  
 @Override  
 **public void** onCheckedChanged(CompoundButton compoundButton, **boolean** isChecked) {  
 String toastMessage;  
 **if**(isChecked){  
  
 **long** triggerTime = SystemClock.*elapsedRealtime*()  
 + AlarmManager.***INTERVAL\_FIFTEEN\_MINUTES***;  
  
 **long** repeatInterval = AlarmManager.***INTERVAL\_FIFTEEN\_MINUTES***;  
  
 *//If the Toggle is turned on, set the repeating alarm with a 15 minute interval* alarmManager.setInexactRepeating(AlarmManager.***ELAPSED\_REALTIME\_WAKEUP***,  
 triggerTime, repeatInterval, notifyPendingIntent);  
  
 *//Set the toast message for the "on" case* toastMessage = getString(R.string.***alarm\_on\_toast***);  
 } **else** {  
 *//Cancel the alarm and notification if the alarm is turned off* alarmManager.cancel(notifyPendingIntent);  
 **mNotificationManager**.cancelAll();  
  
 *//Set the toast message for the "off" case* toastMessage = getString(R.string.***alarm\_off\_toast***);  
 }  
  
 *//Show a toast to say the alarm is turned on or off* Toast.*makeText*(MainActivity.**this**, toastMessage, Toast.***LENGTH\_SHORT***)  
 .show();  
 }  
 });  
  
  
 }  
}

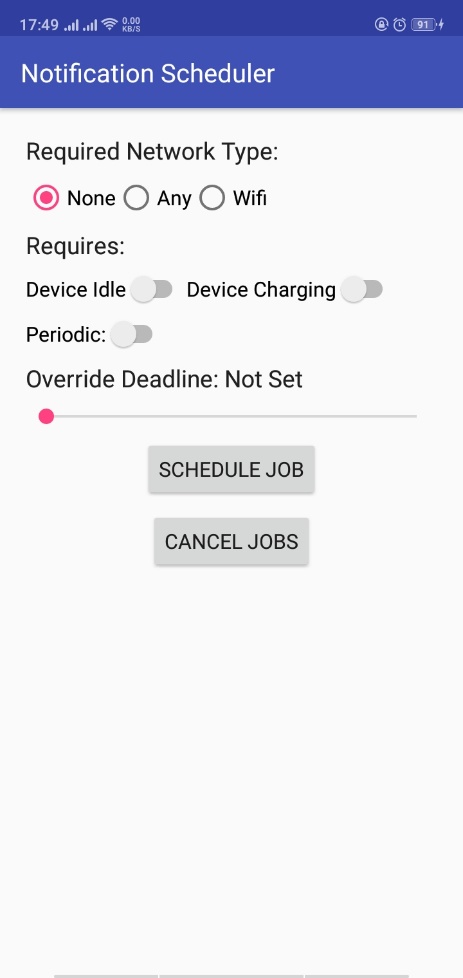
**AlarmReceiver.java**

**package** com.example.android.standup;  
  
**import** android.app.NotificationManager;  
**import** android.app.PendingIntent;  
**import** android.content.BroadcastReceiver;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.support.v4.app.NotificationCompat;  
  
**public class** AlarmReceiver **extends** BroadcastReceiver {  
  
 **private static final int *NOTIFICATION\_ID*** = 0;  
  
  
 **public** AlarmReceiver() {  
 }  
  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
 NotificationManager notificationManager = (NotificationManager)  
 context.getSystemService(Context.***NOTIFICATION\_SERVICE***);  
  
 *//Create the content intent for the notification, which launches 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)  
 .setSmallIcon(R.drawable.***ic\_stand\_up***)  
 .setContentTitle(context.getString(R.string.***notification\_title***))  
 .setContentText(context.getString(R.string.***notification\_text***))  
 .setContentIntent(contentPendingIntent)  
 .setAutoCancel(**true**)  
 .setPriority(NotificationCompat.***PRIORITY\_HIGH***)  
 .setDefaults(NotificationCompat.***DEFAULT\_ALL***);  
  
 *//Deliver the notification* notificationManager.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="com.example.android.standup.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\_centerHorizontal="true"  
 android:layout\_centerVertical="true"** />  
  
  
</**RelativeLayout**>

**Notification scheduler**

****

**String.xml**

<**resources**>  
 <**string name="app\_name"**>Notification Scheduler</**string**>  
 <**string name="periodic\_interval"**>"Periodic Interval: "</**string**>  
 <**string name="override\_deadline"**>"Override Deadline: "</**string**>  
 <**string name="seekbar\_label"**>%1$d s</**string**>  
 <**string name="not\_set"**>Not Set</**string**>  
 <**string name="no\_interval\_toast"**>Please set a periodic interval</**string**>  
 <**string name="job\_scheduled"**>Job Scheduled</**string**>  
 <**string name="jobs\_canceled"**>Jobs Canceled</**string**>  
 <**string name="job\_service"**>Job Service</**string**>  
 <**string name="job\_running"**>Your Job is running!</**string**>  
 <**string name="required\_network\_type"**>Required Network Type:</**string**>  
 <**string name="no\_network"**>None</**string**>  
 <**string name="any\_network"**>Any</**string**>  
 <**string name="wifi\_network"**>Wifi</**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="schedule\_job"**>Schedule Job</**string**>  
 <**string name="cancel\_jobs"**>Cancel Jobs</**string**>  
 <**string name="no\_constraint\_toast"**>Please set at least one constraint</**string**>  
</**resources**>

**MainActivity.java**

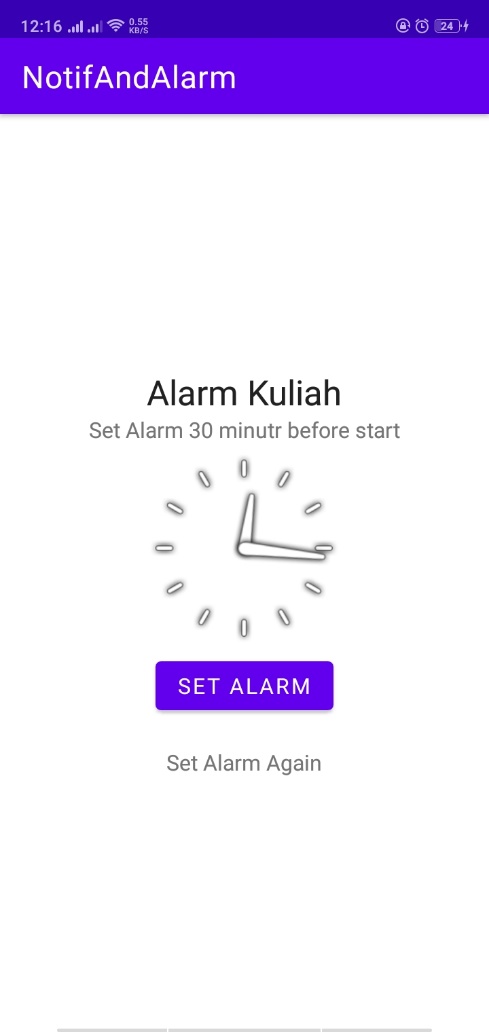
**package** com.example.android.notificationscheduler;  
  
**import** android.app.job.JobInfo;  
**import** android.app.job.JobScheduler;  
**import** android.content.ComponentName;  
**import** android.os.Bundle;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.view.View;  
**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 single deadlines* **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();  
 }  
 }  
  
 */\*\*  
 \* onClick method for cancelling all existing jobs  
 \*/* **private void** cancelJobs() {  
 **if** (**mScheduler** != **null**){  
 **mScheduler**.cancelAll();  
 **mScheduler** = **null**;  
 Toast.*makeText*(**this**, R.string.***jobs\_canceled***, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
}

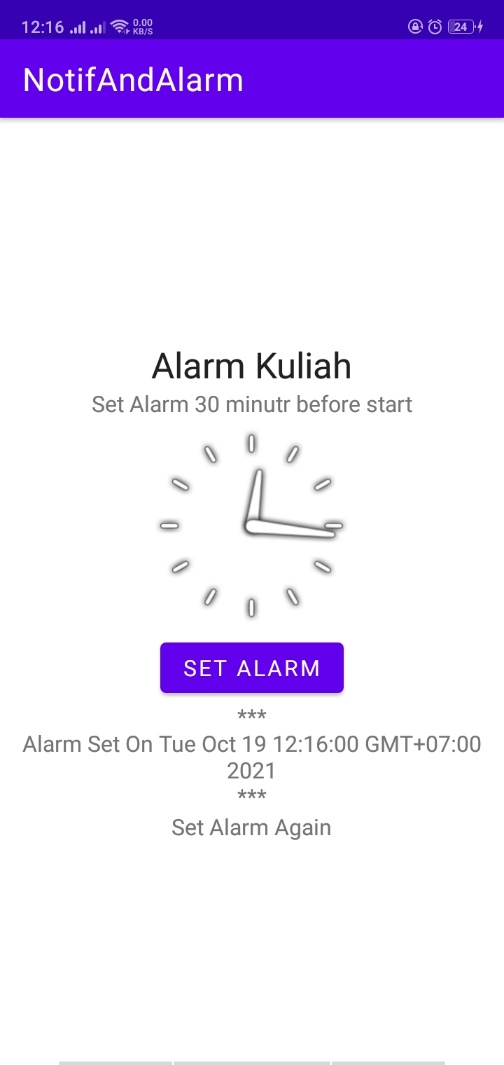
**NotificationJobService.java**

**package** com.example.android.notificationscheduler;  
  
**import** android.app.NotificationManager;  
**import** android.app.PendingIntent;  
**import** android.app.job.JobParameters;  
**import** android.app.job.JobService;  
**import** android.content.Intent;  
**import** android.support.v4.app.NotificationCompat;  
  
**public class** NotificationJobService **extends** JobService {  
   
 @Override  
 **public boolean** onStartJob(JobParameters jobParameters) {  
  
 *//Set up the notification content intent to launch the app when clicked* 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.***ic\_job\_running***)  
 .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="com.example.android.notificationscheduler.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**>

**Notif and alarm**

****

**String.xml**

<resources>

    <string *name*="app\_name">NotifAndAlarm</string>

    <string *name*="head">Alarm Kuliah</string>

    <string *name*="set\_alarm">Set Alarm 30 minute before start</string>

    <string *name*="alarm\_set">Set Alarm</string>

    <string *name*="content">Ayo Kuliah</string>

    <string *name*="alarm\_again">Set Alarm Again</string>

</resources>

**MainActivity.java**

*package* *com.example.notifandalarm*;

*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 Title");

*timePickerDialog*.show();

            }

        });

    }

*TimePickerDialog*.*OnTimeSetListener* onTimeSetListener = new TimePickerDialog.OnTimeSetListener() {

        @*Override*

*public* *void* onTimeSet(*TimePicker* timePicker, *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(), *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(),repeatInterval,pi);

        }

    };

}

**Alarmreceiver.java**

*package* *com.example.notifandalarm*;

*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.widget.Toast*;

*import* *androidx.core.app.NotificationCompat*;

*public* *class* AlarmReceiver *extends* BroadcastReceiver {

*MediaPlayer* mp;

*private* *static* *final* *int* NOTIFICATION\_ID = 0;

    @*Override*

*public* *void* onReceive(*Context* context, *Intent* intent) {

*Toast*.makeText(context, "Alarm Aktif", *Toast*.*LENGTH\_SHORT*).show();

*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)

                .setContentTitle(*context*.getString(*R*.*string*.*head*))

                .setContentText(*context*.getString(*R*.*string*.*content*))

                .setContentIntent(pi)

                .setAutoCancel(true)

                .setPriority(*NotificationCompat*.*PRIORITY\_HIGH*)

                .setDefaults(*NotificationCompat*.*DEFAULT\_ALL*);

*notifMgr*.notify(NOTIFICATION\_ID, *build*.build());

    }

}

**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:layout\_marginVertical*="10dp"

*android:layout\_marginHorizontal*="10dp"

*android:gravity*="center"

*tools:context*=".MainActivity">

    <TextView

*android:id*="@+id/tv1"

*android:layout\_width*="wrap\_content"

*android:layout\_height*="wrap\_content"

*android:text*="@string/head"

*android:textAppearance*="@style/TextAppearance.AppCompat.Large"/>

    <TextView

*android:id*="@+id/tv2"

*android:layout\_width*="wrap\_content"

*android:layout\_height*="wrap\_content"

*android:text*="@string/set\_alarm"

*android:textAppearance*="@style/TextAppearance.AppCompat.Small"/>

    <AnalogClock

*android:id*="@+id/analog"

*android:layout\_width*="wrap\_content"

*android:layout\_height*="wrap\_content"/>

    <Button

*android:id*="@+id/btnStart"

*android:layout\_width*="wrap\_content"

*android:layout\_height*="wrap\_content"

*android:text*="@string/alarm\_set"/>

    <TextView

*android:id*="@+id/alarm"

*android:layout\_width*="wrap\_content"

*android:layout\_height*="wrap\_content"

*android:gravity*="center"/>

    <TextView

*android:layout\_width*="wrap\_content"

*android:layout\_height*="wrap\_content"

*android:text*="@string/alarm\_again"

*android:id*="@+id/tv3"

*android:textAppearance*="@style/TextAppearance.AppCompat.Small"/>

</LinearLayout>