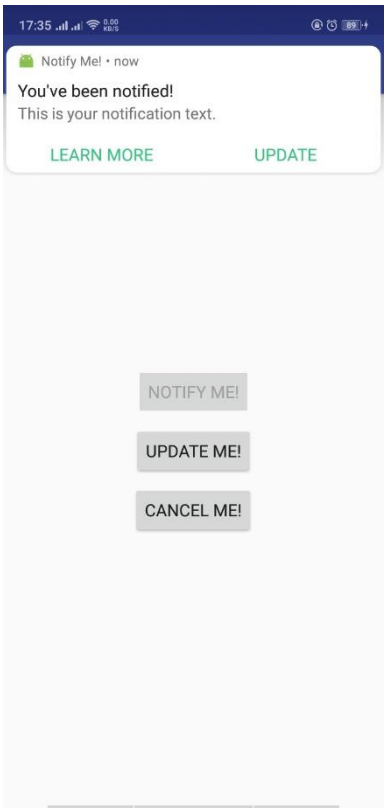
Nama : **Bayu Prasetya Adji Sugiyarto**

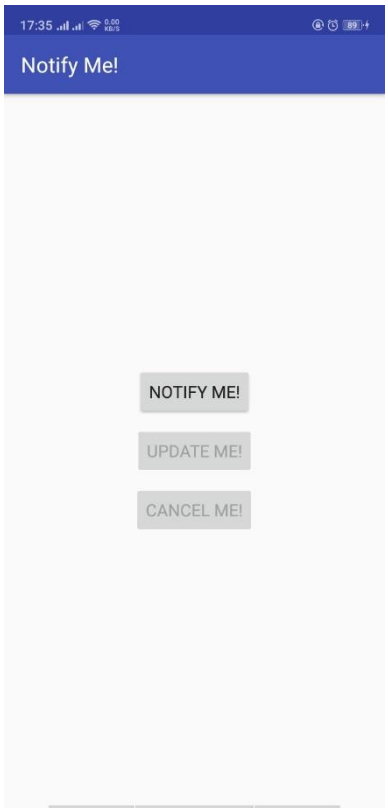
NIM : **A11.2019.11688**

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();
            }
        });


    }
}
```

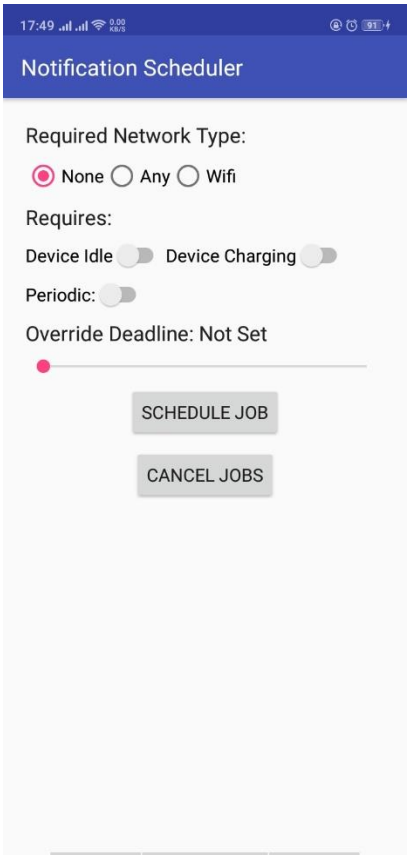
AlarmReceiver.java

```
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;
    }
}
```

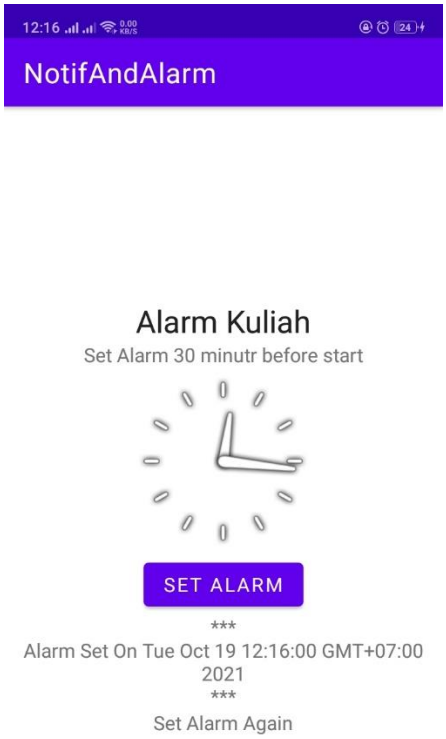
Activity_main.xml

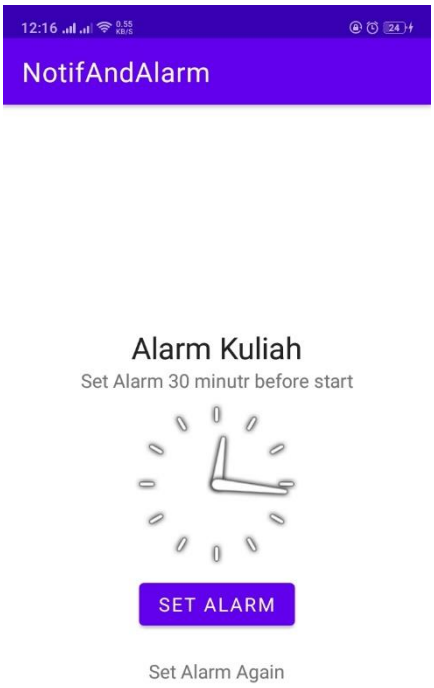
```
<?xml version="1.0" encoding="utf-8"?>

<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.getTimeI
nMillis(),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>
```