

Modul 7 – Alarm Manager

Tujuan

Kali ini kita akan membuat sebuah proyek sederhana dengan skenario seperti berikut.

1. Aplikasi akan memiliki kemampuan untuk melakukan *one-time-execution alarm* dan alarm yang berulang berdasarkan waktu yang ditentukan.
2. Aplikasi akan menyimpan beberapa nilai untuk kedua tipe alarm tersebut.
3. Aplikasi akan menampilkan notifikasi dengan suara dan getar ketika alarm dijalankan.

Codelab

1. Buat proyek baru dengan nama **MyAlarmManager** yang berjenis **Empty Activity** dan dengan konfigurasi *default* tentunya.
2. Pada **activity_main.xml** sesuaikan kodenya menjadi seperti berikut.

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:tools="http://schemas.android.com/tools"
4.     android:id="@+id/activity_main"
5.     android:layout_width="match_parent"
6.     android:layout_height="match_parent"
7.     android:paddingBottom="@dimen/activity_vertical_margin"
8.     android:paddingLeft="@dimen/activity_horizontal_margin"
9.     android:paddingRight="@dimen/activity_horizontal_margin"
10.    android:paddingTop="@dimen/activity_vertical_margin"
11.    android:orientation="vertical"
12.    tools:context="com.dicoding.myalarmmanager.MainActivity">
13.    <TextView
14.        android:layout_width="match_parent"
15.        android:layout_height="wrap_content"
16.        android:text="One Time Alarm"
17.        android:layout_marginBottom="@dimen/activity_vertical_margin"
18.        android:textSize="16sp"
19.        android:textStyle="bold"/>
20.    <LinearLayout
21.        android:layout_width="match_parent"
22.        android:layout_height="wrap_content"
23.        android:orientation="horizontal">
24.        <Button
25.            android:text="..."
26.            android:layout_width="50dp"
27.            android:layout_height="wrap_content"
28.            android:id="@+id/btn_one_time_alarm_date" />
```

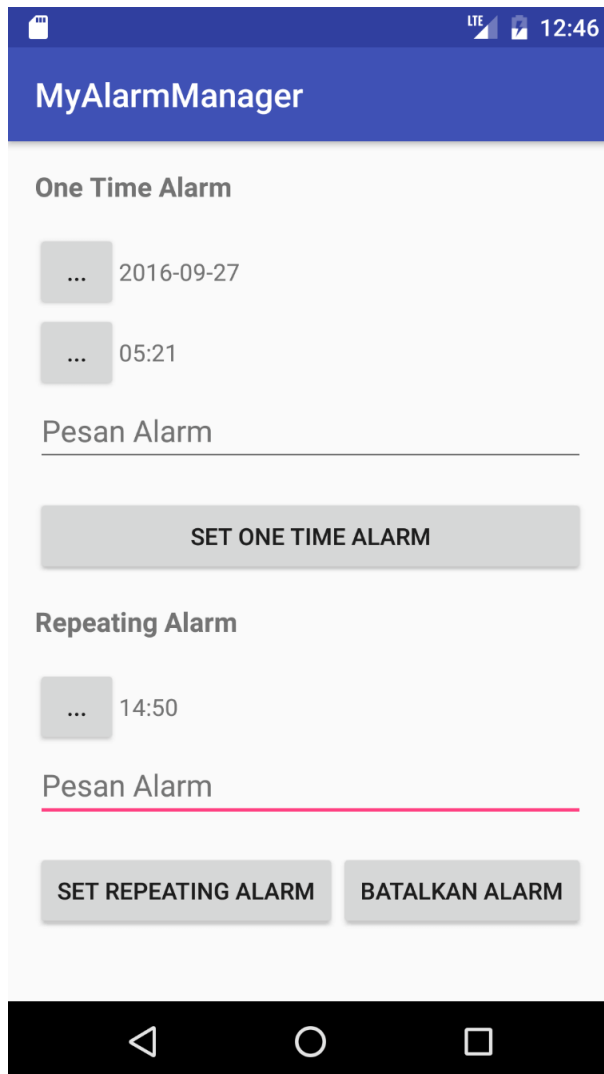
```

29.         <TextView
30.             android:id="@+id/tv_one_time_alarm_date"
31.             android:layout_width="wrap_content"
32.             android:layout_height="wrap_content"
33.             android:text="Tanggal Alarm"
34.             android:layout_marginBottom="@dimen/activity_vertical_ma
rgin"/>
35.     </LinearLayout>
36.     <LinearLayout
37.         android:layout_width="match_parent"
38.         android:layout_height="wrap_content"
39.         android:orientation="horizontal">
40.         <Button
41.             android:text="..."
42.             android:layout_width="50dp"
43.             android:layout_height="wrap_content"
44.             android:id="@+id/btn_one_time_alarm_time" />
45.
46.         <TextView
47.             android:id="@+id/tv_one_time_alarm_time"
48.             android:layout_width="wrap_content"
49.             android:layout_height="wrap_content"
50.             android:text="Waktu Alarm"
51.             android:layout_marginBottom="@dimen/activity_vertical_ma
rgin"/>
52.     </LinearLayout>
53.     <EditText
54.         android:id="@+id/edt_one_time_alarm_message"
55.         android:layout_width="match_parent"
56.         android:layout_height="wrap_content"
57.         android:hint="Pesan Alarm"
58.         android:layout_marginBottom="@dimen/activity_vertical_margin
"/>
59.     <Button
60.         android:id="@+id/btn_set_one_time_alarm"
61.         android:layout_width="match_parent"
62.         android:layout_height="wrap_content"
63.         android:text="Set One Time Alarm"
64.         android:layout_marginBottom="@dimen/activity_vertical_margin
"/>
65.     <TextView
66.         android:layout_width="match_parent"
67.         android:layout_height="wrap_content"
68.         android:text="Repeating Alarm"
69.         android:layout_marginBottom="@dimen/activity_vertical_margin
"
70.         android:textSize="16sp"
71.         android:textStyle="bold"/>
72.     <LinearLayout
73.         android:layout_width="match_parent"
74.         android:layout_height="wrap_content"
75.         android:orientation="horizontal">
76.         <Button
77.             android:text="..."
78.             android:layout_width="50dp"
79.             android:layout_height="wrap_content"
80.             android:id="@+id/btn_repeating_time_alarm_time" />
81.         <TextView
82.             android:id="@+id/tv_repeating_alarm_time"

```

```
83.         android:layout_width="wrap_content"
84.         android:layout_height="wrap_content"
85.         android:text="Waktu Alarm"
86.         android:layout_marginBottom="@dimen/activity_vertical_ma
    rgin"/>
87.     </LinearLayout>
88.     <EditText
89.         android:id="@+id/tv_repeating_alarm_message"
90.         android:layout_width="match_parent"
91.         android:layout_height="wrap_content"
92.         android:hint="Pesan Alarm"
93.         android:layout_marginBottom="@dimen/activity_vertical_margin"
    "/>
94.     <LinearLayout
95.         android:layout_width="match_parent"
96.         android:layout_height="wrap_content"
97.         android:orientation="horizontal">
98.         <Button
99.             android:id="@+id/btn_repeating_time_alarm"
100.            android:layout_width="wrap_content"
101.            android:layout_height="wrap_content"
102.            android:text="Set Repeating Alarm"
103.            android:layout_weight="0.5"
104.        />
105.        <Button
106.            android:id="@+id/btn_cancel_repeating_alarm"
107.            android:layout_width="wrap_content"
108.            android:layout_height="wrap_content"
109.            android:text="Batalikan Alarm"
110.            android:layout_weight="0.5"
111.        />
112.    </LinearLayout>
113. </LinearLayout>
```

Sehingga tampilan aplikasi dari `activity_main.xml` ketika dijalankan seperti berikut :



3. Setelah itu kita akan membuat sebuah mekanisme penyimpanan sederhana dengan menggunakan **SharedPreference**. Untuk itu, mari kita membuat sebuah kelas java baru dengan nama **AlarmPreference**.

Caranya, klik kanan pada package project yang ada saat ini → **new** → **Java Class**. Ketika sudah selesai, lengkapi kodenya menjadi seperti berikut.

```

1. public class AlarmPreference {
2.     private final String PREF_NAME = "AlarmPreference";
3.     private final String KEY_ONE_TIME_DATE = "oneTimDate";
4.     private final String KEY_ONE_TIME_TIME = "oneTimeTime";
5.     private final String KEY_ONE_TIME_MESSAGE = "oneTimeMessage";
6.     private final String KEY_REPEATING_TIME = "repeatingTime";
7.     private final String KEY_REPEATING_MESSAGE = "repeatingMessage";
8.     private SharedPreferences mSharedPreferences;
9.     private SharedPreferences.Editor editor;
10.    public AlarmPreference(Context context){
11.        mSharedPreferences = context.getSharedPreferences(PREF_NAME,
12.            Context.MODE_PRIVATE);
13.        editor = mSharedPreferences.edit();
14.    }
15.    public void setOneTimeDate(String date){
16.        editor.putString(KEY_ONE_TIME_DATE, date);
17.        editor.commit();
18.    }
19.    public void setOneTimeTime(String time){
20.        editor.putString(KEY_ONE_TIME_TIME, time);
21.        editor.commit();
22.    }
23.    public void setOneTimeMessage(String message){
24.        editor.putString(KEY_ONE_TIME_MESSAGE, message);
25.        editor.commit();
26.    }
27.    public void setRepeatingTime(String time){
28.        editor.putString(KEY_REPEATING_TIME, time);
29.        editor.commit();
30.    }
31.    public void setRepeatingMessage(String message){
32.        editor.putString(KEY_REPEATING_MESSAGE, message);
33.        editor.commit();
34.    }
35. }

```

```

17.     }
18.     public String getOneTimeDate(){
19.         return mSharedPreferences.getString(KEY_ONE_TIME_DATE, null);
20.     }
21.     public void setOneTimeTime(String time){
22.         editor.putString(KEY_ONE_TIME_TIME, time);
23.         editor.commit();
24.     }
25.     public String getOneTimeTime(){
26.         return mSharedPreferences.getString(KEY_ONE_TIME_TIME, null);
27.     }
28.     public void setOneTimeMessage(String message){
29.         editor.putString(KEY_ONE_TIME_MESSAGE, message);
30.         editor.commit();
31.     }
32.     public String getOneTimeMessage(){
33.         return mSharedPreferences.getString(KEY_ONE_TIME_MESSAGE, nul
34.     1);
35.     }
36.     public void setRepeatingTime(String time){
37.         editor.putString(KEY_REPEATING_TIME, time);
38.         editor.commit();
39.     }
40.     public String getRepeatingTime(){
41.         return mSharedPreferences.getString(KEY_REPEATING_TIME, null)
42.     ;
43.     }
44.     public void setRepeatingMessage(String message){
45.         editor.putString(KEY_REPEATING_MESSAGE, message);
46.         editor.commit();
47.     }
48.     public String getRepeatingMessage(){
49.         return mSharedPreferences.getString(KEY_REPEATING_MESSAGE, nu
50.     11);
51.     }
52.     public void clear(){
53.         editor.clear();
54.         editor.commit();
55.     }
56. }

```

4. Setelah itu, buat kelas **BroadcastReceiver** bernama **AlarmReceiver** dan lengkapi kodenya seperti berikut.

```

1. public class AlarmReceiver extends BroadcastReceiver {
2.     public static final String TYPE_ONE_TIME = "OneTimeAlarm";
3.     public static final String TYPE_REPEATING = "RepeatingAlarm";
4.     public static final String EXTRA_MESSAGE = "message";
5.     public static final String EXTRA_TYPE = "type";
6.     private final int NOTIF_ID_ONETIME = 100;
7.     private final int NOTIF_ID_REPEATING = 101;
8.
9.     public AlarmReceiver() {
10.    }
11.    @Override
12.    public void onReceive(Context context, Intent intent) {
13.        String type = intent.getStringExtra(EXTRA_TYPE);

```

```

14.     String message = intent.getStringExtra(EXTRA_MESSAGE);
15.     String title = type.equalsIgnoreCase(TYPE_ONE_TIME) ? "One Time Alarm" : "Repeating Alarm";
16.     int notifId = type.equalsIgnoreCase(TYPE_ONE_TIME) ? NOTIF_ID_ONETIME : NOTIF_ID_REPEATING;
17.     showAlarmNotification(context, title, message, notifId);
18. }
19. private void showAlarmNotification(Context context, String title, String message, int notifId){
20.     NotificationManager notificationManagerCompat = (NotificationManager) context.getSystemService(Context.NOTIFICATION_SERVICE);
21.     Uri alarmSound = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
22.     NotificationCompat.Builder builder = new NotificationCompat.Builder(context)
23.         .setSmallIcon(R.drawable.ic_access_alarm_black_36dp)
24.         .setContentTitle(title)
25.         .setContentText(message)
26.         .setColor(ContextCompat.getColor(context, android.R.color.transparent))
27.         .setVibrate(new long[]{1000, 1000, 1000, 1000, 1000})
28.         .setSound(alarmSound);
29.     notificationManagerCompat.notify(notifId, builder.build());
30. }
31. public void setOneTimeAlarm(Context context, String type, String date, String time, String message){
32.     AlarmManager alarmManager = (AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
33.     Intent intent = new Intent(context, AlarmReceiver.class);
34.     intent.putExtra(EXTRA_MESSAGE, message);
35.     intent.putExtra(EXTRA_TYPE, type);
36.     String dateArray[] = date.split("-");
37.     String timeArray[] = time.split(":");
38.     Calendar calendar = Calendar.getInstance();
39.     calendar.set(Calendar.YEAR, Integer.parseInt(dateArray[0]));
40.     calendar.set(Calendar.MONTH, Integer.parseInt(dateArray[1])-1);
41.     calendar.set(Calendar.DAY_OF_MONTH, Integer.parseInt(dateArray[2]));
42.     calendar.set(Calendar.HOUR_OF_DAY, Integer.parseInt(timeArray[0]));
43.     calendar.set(Calendar.MINUTE, Integer.parseInt(timeArray[1]));
44.     calendar.set(Calendar.SECOND, 0);
45.     int requestCode = NOTIF_ID_ONETIME;
46.     PendingIntent pendingIntent = PendingIntent.getBroadcast(context, requestCode, intent, 0);
47.     alarmManager.set(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(), pendingIntent);
48.     Toast.makeText(context, "One time alarm set up", Toast.LENGTH_SHORT).show();
49. }
50. public void cancelAlarm(Context context, String type){
51.     AlarmManager alarmManager = (AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
52.     Intent intent = new Intent(context, AlarmReceiver.class);
53.     int requestCode = type.equalsIgnoreCase(TYPE_ONE_TIME) ? NOTIF_ID_ONETIME : NOTIF_ID_REPEATING;

```

```

54.         PendingIntent pendingIntent = PendingIntent.getBroadcast(cont
ext, requestCode, intent, 0);
55.         alarmManager.cancel(pendingIntent);
56.     }
57. }

```

Untuk keperluan ikon alarm, kami menggunakan ikon dari material design spec yang dapat dicari di [halaman ini](#), dibagian kategori peranti. Setelah diunduh, masukkan ikon ke dalam folder **res** → **drawable**.

- Setelah kita membuat sebuah **AlarmReceiver** dan menyesuaikan kode pada **activity_main.xml**, sekarang saatnya kita melengkapi **MainActivity** kodenya sehingga menjadi seperti ini :

```

1. public class MainActivity extends AppCompatActivity implements View.
   OnClickListener{
2.     private TextView tvOneTimeDate, tvOneTimeTime ;
3.     private TextView tvRepeatingTime;
4.     private EditText edtOneTimeMessage , edtRepeatingMessage;
5.     private Button btnOneTimeDate, btnOneTimeTime, btnOneTime ,btnRe
      peatingTime ,btnRepeating,  btnCancelRepeatingAlarm;
6.
7.     private Calendar calOneTimeDate, calOneTimeTime , calRepeatTimeT
      ime;
8.
9.     private AlarmReceiver alarmReceiver;
10.    private AlarmPreference alarmPreference;
11.
12.    @Override
13.    protected void onCreate(Bundle savedInstanceState) {
14.        super.onCreate(savedInstanceState);
15.        setContentView(R.layout.activity_main);
16.
17.        tvOneTimeDate = (TextView)findViewById(R.id.tv_one_time_alar
      m_date);
18.        tvOneTimeTime = (TextView)findViewById(R.id.tv_one_time_alar
      m_time);
19.        edtOneTimeMessage = (EditText)findViewById(R.id.edt_one_time
      _alarm_message);
20.        btnOneTimeDate = (Button)findViewById(R.id.btn_one_time_alar
      m_date);
21.        btnOneTimeTime = (Button)findViewById(R.id.btn_one_time_alar
      m_time);
22.        btnOneTime = (Button)findViewById(R.id.btn_set_one_time_alar
      m);
23.
24.        tvRepeatingTime = (TextView)findViewById(R.id.tv_repeating_a
      larm_time);
25.        edtRepeatingMessage = (EditText)findViewById(R.id.edt_repeat
      ing_alarm_message);
26.        btnRepeatingTime = (Button)findViewById(R.id.btn_repeating_t
      ime_alarm_time);
27.        btnRepeating = (Button)findViewById(R.id.btn_repeating_time_
      alarm);
28.
29.        btnCancelRepeatingAlarm = (Button)findViewById(R.id.btn_canc
      el_repeating_alarm);
30.
31.        btnOneTimeDate.setOnClickListener(this);

```



```

32.         btnOneTimeTime.setOnClickListener(this);
33.         btnOneTime.setOnClickListener(this);
34.         btnRepeatingTime.setOnClickListener(this);
35.         btnRepeating.setOnClickListener(this);
36.         btnCancelRepeatingAlarm.setOnClickListener(this);
37.
38.         calOneTimeDate = Calendar.getInstance();
39.         calOneTimeTime = Calendar.getInstance();
40.         calRepeatTimeTime = Calendar.getInstance();
41.
42.
43.         alarmPreference = new AlarmPreference(this);
44.         alarmReceiver = new AlarmReceiver();
45.
46.         if (!TextUtils.isEmpty(alarmPreference.getOneTimeDate())){
47.             setOneTimeText();
48.         }
49.
50.     }
51.     @Override
52.     public void onClick(View v) {
53.         if (v.getId() == R.id.btn_one_time_alarm_date){
54.             final Calendar currentDate = Calendar.getInstance();
55.             new DatePickerDialog(this, new DatePickerDialog.OnDateSe
tListener() {
56.                 @Override
57.                 public void onDateSet(DatePicker view, int year, int
monthOfYear, int dayOfMonth) {
58.                     calOneTimeDate.set(year, monthOfYear, dayOfMon
th
59. );
60.                     SimpleDateFormat dateFormat = new SimpleDateFormat
at("yyyy-MM-dd");
61.                     tvOneTimeDate.setText(dateFormat.format(calOneTi
meDate.getTime()));
62.                 }, currentDate.get(Calendar.YEAR), currentDate.get(Cal
endar.MONTH), currentDate.get(Calendar.DATE)).show();
63.             }
64.             else if (v.getId() == R.id.btn_one_time_alarm_time){
65.                 final Calendar currentDate = Calendar.getInstance();
66.                 new TimePickerDialog(this, new TimePickerDialog.OnTimeSe
tListener() {
67.                     @Override
68.                     public void onTimeSet(TimePicker view, int hourOfDay
, int minute) {
69.                         calOneTimeTime.set(Calendar.HOUR_OF_DAY, hourOfD
ay);
70.                         calOneTimeTime.set(Calendar.MINUTE, minute);
71.                         SimpleDateFormat dateFormat = new SimpleDateFormat
at("HH:mm");
72.                         tvOneTimeTime.setText(dateFormat.format(calOneTi
meTime.getTime()));
73.                         //Log.v(TAG, "The choosen one " + date.getTime()
);
74.                     }
75.                 }, currentDate.get(Calendar.HOUR_OF_DAY), currentDate.ge
t(Calendar.MINUTE), true).show();
76.             }
77.         }

```



```
78.         else if (v.getId() == R.id.btn_set_one_time_alarm){
79.             SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy
-MM-dd");
80.
81.             String oneTimeDate = dateFormat.format(calOneTimeDate.ge
tTime());
82.
83.             SimpleDateFormat timeFormat = new SimpleDateFormat("HH:m
m");
84.
85.             String oneTimeTime = timeFormat.format(calOneTimeTime.ge
tTime());
86.             String oneTimeMessage = edtOneTimeMessage.getText().toSt
ring();
87.
88.             alarmPreference.setOneTimeDate(oneTimeDate);
89.             alarmPreference.setOneTimeMessage(oneTimeMessage);
90.             alarmPreference.setOneTimeTime(oneTimeTime);
91.
92.             alarmReceiver.setOneTimeAlarm(this, AlarmReceiver.TYPE_O
NE_TIME,
93.                 alarmPreference.getOneTimeDate(),
94.                 alarmPreference.getOneTimeTime(),
95.                 alarmPreference.getOneTimeMessage());
96.         }
97.
98.     }
99.
100.    private void setOneTimeText() {
101.        tvOneTimeTime.setText(alarmPreference.getOneTimeTime());
102.        tvOneTimeDate.setText(alarmPreference.getOneTimeDate());
103.        edtOneTimeMessage.setText(alarmPreference.getOneTimeMessa
ge());
104.    }
105. }
```

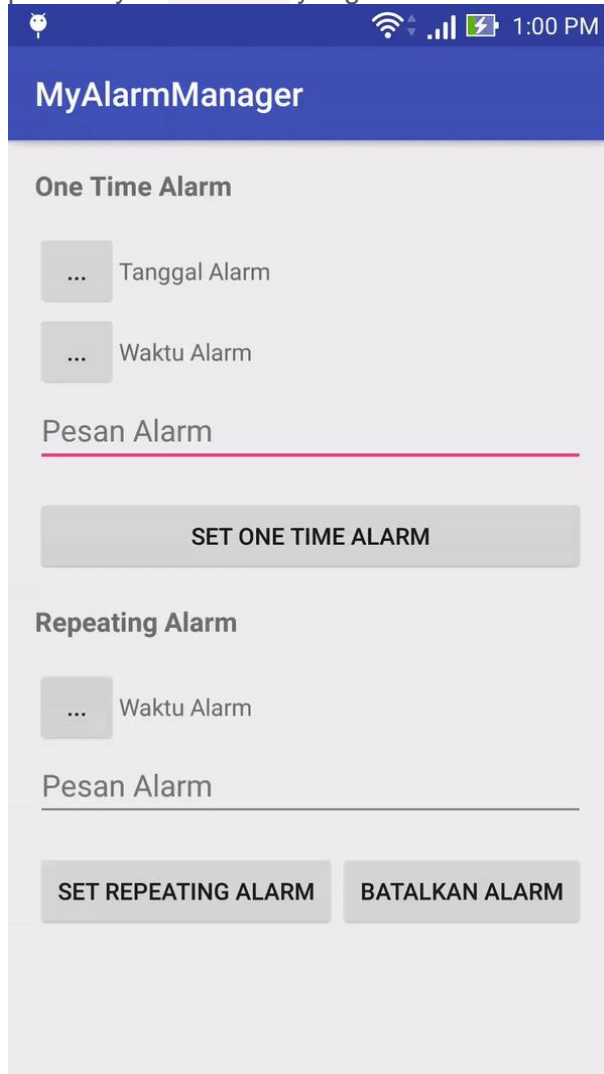
6. Jangan lupa untuk menambahkan permission pada **AndroidManifest.xml** beberapa baris berikut.

```
1. <uses-permission android:name="android.permission.VIBRATE" />
2. <uses-permission android:name="android.permission.WAKE_LOCK" />
```

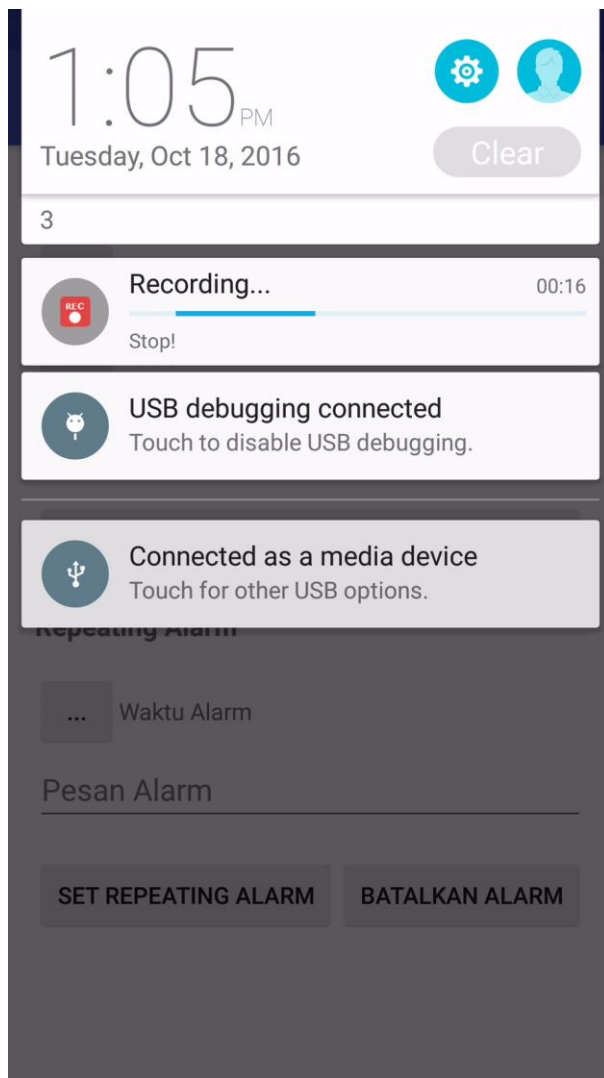
Ketika mengimport komponen **Calendar**, pastikan Anda menggunakan **java.util.calendar**. Jangan sampai keliru dengan komponen calendar lainnya yang baru ditambahkan pada API 24.

7. Jika sudah selesai, silakan jalankan aplikasi Anda. Dengan menekan tombol yang ada pada sebelah **TextView** tanggal dan waktu, Anda dapat mengganti tanggal dan waktu yang diinginkan. Isikan juga pesan alarm yang diinginkan. Setelah tanggal, waktu dan pesan telah Anda isi, lanjutkan dengan klik tombol **SET ONE TIME ALARM** dan silakan tunggu sampai waktu yang ditentukan, sehingga alarm akan dijalankan. Jika sukses, maka waktu yang kita set di awal akan tampil

pada obyek `TextView` yang telah disediakan.



Ketika waktunya sudah sesuai, maka alarm akan dijalankan dan akan menampilkan notifikasi dengan getaran dan suara seperti berikut.



Keren! Sekarang Anda sudah bisa mengimplementasikan *one-time-alarm* untuk tanggal dan jam yang telah ditentukan. Fungsi ini bisa berguna untuk beragam fitur lain. Misalnya jika Anda membuat aplikasi tentang *event*, pasti fungsi ini akan sangat berguna sebagai mekanisme pengingat.

Kita lanjutkan ngoding untuk latihan alarm manager. Kali ini kita akan membuat latihan sederhana tentang *repeating-alarm*.

1. Pada `AlarmReceiver` silakan buat metode `setRepeatingAlarm()` pada baris berikut.

```

1. public void setRepeatingAlarm(Context context, String type, String time, String message){
2.     AlarmManager alarmManager = (AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
3.     Intent intent = new Intent(context, AlarmReceiver.class);
4.     intent.putExtra(EXTRA_MESSAGE, message);
5.     intent.putExtra(EXTRA_TYPE, type);
6.     String timeArray[] = time.split(":");
7.     Calendar calendar = Calendar.getInstance();
8.     calendar.set(Calendar.HOUR_OF_DAY, Integer.parseInt(timeArray[0]));
9.     calendar.set(Calendar.MINUTE, Integer.parseInt(timeArray[1]));
10.    calendar.set(Calendar.SECOND, 0);
11.    int requestCode = NOTIF_ID_ONETIME;
12.    PendingIntent pendingIntent = PendingIntent.getBroadcast(context, requestCode, intent, 0);
13.    alarmManager.setInexactRepeating(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(), AlarmManager.INTERVAL_DAY, pendingIntent);
14.    Toast.makeText(context, "Repeating alarm set up", Toast.LENGTH_SHORT).show();
15.}
  
```

2. Sehingga kode `AlarmManager` kita saat ini menjadi seperti ini.

```

1. public class AlarmReceiver extends BroadcastReceiver {
2.     public static final String TYPE_ONE_TIME = "OneTimeAlarm";
3.     public static final String TYPE_REPEATING = "RepeatingAlarm";
4.     public static final String EXTRA_MESSAGE = "message";
5.     public static final String EXTRA_TYPE = "type";
6.     private final int NOTIF_ID_ONETIME = 100;
7.     private final int NOTIF_ID_REPEATING = 101;
8.     public AlarmReceiver() {
9.     }
10.    @Override
11.    public void onReceive(Context context, Intent intent) {
12.        String type = intent.getStringExtra(EXTRA_TYPE);
13.        String message = intent.getStringExtra(EXTRA_MESSAGE);
14.        String title = type.equalsIgnoreCase(TYPE_ONE_TIME) ? "One Time Alarm" : "Repeating Alarm";
15.        int notifId = type.equalsIgnoreCase(TYPE_ONE_TIME) ? NOTIF_ID_ONETIME : NOTIF_ID_REPEATING;
16.        showAlarmNotification(context, title, message, notifId);
17.    }
18.    private void showAlarmNotification(Context context, String title, String message, int notifId){
19.        NotificationManager notificationManagerCompat = (NotificationManager) context.getSystemService(Context.NOTIFICATION_SERVICE);
20.        Uri alarmSound = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
  
```

```

21.         NotificationCompat.Builder builder = new NotificationCompat.
           Builder(context)
22.             .setSmallIcon(R.drawable.ic_access_alarm_black_36dp)
23.             .setContentTitle(title)
24.             .setContentText(message)
25.             .setColor(ContextCompat.getColor(context, android.R.
           color.transparent))
26.             .setVibrate(new long[]{1000, 1000, 1000, 1000, 1000}
           )
27.             .setSound(alarmSound);
28.         notificationManagerCompat.notify(notifId, builder.build());
29.     }
30.     public void setOneTimeAlarm(Context context, String type, String
           date, String time, String message){
31.         AlarmManager alarmManager = (AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
32.         Intent intent = new Intent(context, AlarmReceiver.class);
33.         intent.putExtra(EXTRA_MESSAGE, message);
34.         intent.putExtra(EXTRA_TYPE, type);
35.         String dateArray[] = date.split("-");
36.         String timeArray[] = time.split(":");
37.         Calendar calendar = Calendar.getInstance();
38.         calendar.set(Calendar.YEAR, Integer.parseInt(dateArray[0]));
39.         calendar.set(Calendar.MONTH, Integer.parseInt(dateArray[1])-
           1);
40.         calendar.set(Calendar.DAY_OF_MONTH, Integer.parseInt(dateArr
           ay[2]));
41.         calendar.set(Calendar.HOUR_OF_DAY, Integer.parseInt(timeArra
           y[0]));
42.         calendar.set(Calendar.MINUTE, Integer.parseInt(timeArray[1])
           );
43.         calendar.set(Calendar.SECOND, 0);
44.         int requestCode = NOTIF_ID_ONETIME;
45.         PendingIntent pendingIntent = PendingIntent.getBroadcast(co
           ntext, requestCode, intent, 0);
46.         alarmManager.set(AlarmManager.RTC_WAKEUP, calendar.getTimeIn
           Millis(), pendingIntent);
47.         Toast.makeText(context, "One time alarm set up", Toast.LENGT
           H_SHORT).show();
48.     }
49.     public void setRepeatingAlarm(Context context, String type, Stri
           ng time, String message){
50.         AlarmManager alarmManager = (AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
51.         Intent intent = new Intent(context, AlarmReceiver.class);
52.         intent.putExtra(EXTRA_MESSAGE, message);
53.         intent.putExtra(EXTRA_TYPE, type);
54.         String timeArray[] = time.split(":");
55.         Calendar calendar = Calendar.getInstance();
56.         calendar.set(Calendar.HOUR_OF_DAY, Integer.parseInt(timeArra
           y[0]));
57.         calendar.set(Calendar.MINUTE, Integer.parseInt(timeArray[1])
           );
58.         calendar.set(Calendar.SECOND, 0);
59.         int requestCode = NOTIF_ID_REPEATING;
60.         PendingIntent pendingIntent = PendingIntent.getBroadcast(co
           ntext, requestCode, intent, 0);
61.         alarmManager.setInexactRepeating(AlarmManager.RTC_WAKEUP, ca
           lendar.getTimeInMillis(), AlarmManager.INTERVAL_DAY, pendingIntent);

```

```

62.         Toast.makeText(context, "Repeating alarm set up", Toast.LENGTH_SHORT).show();
63.     }
64.     public void cancelAlarm(Context context, String type){
65.         AlarmManager alarmManager = (AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
66.         Intent intent = new Intent(context, AlarmReceiver.class);
67.         int requestCode = type.equalsIgnoreCase(TYPE_ONE_TIME) ? NOTIF_ID_ONETIME : NOTIF_ID_REPEATING;
68.         PendingIntent pendingIntent = PendingIntent.getBroadcast(context, requestCode, intent, 0);
69.         alarmManager.cancel(pendingIntent);
70.     }
71. }
  
```

3. Selanjutnya, pada method `onClick()` di `MainActivity` silakan tambahkan baris berikut :

```

1. else if (v.getId() == R.id.btn_repeating_time_alarm_time){
2.     final Calendar currentDate = Calendar.getInstance();
3.     new TimePickerDialog(this, new TimePickerDialog.OnTimeSetListener() {
4.         @Override
5.         public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
6.             calRepeatTimeTime.set(Calendar.HOUR_OF_DAY, hourOfDay);
7.             calRepeatTimeTime.set(Calendar.MINUTE, minute);
8.             SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm");
9.             tvRepeatingTime.setText(dateFormat.format(calRepeatTimeTime.getTime()));
10.        }
11.    }, currentDate.get(Calendar.HOUR_OF_DAY), currentDate.get(Calendar.MINUTE), true).show();
12. }
13. else if (v.getId() == R.id.btn_repeating_time_alarm){
14.     SimpleDateFormat timeFormat = new SimpleDateFormat("HH:mm");
15.     String repeatTimeTime = timeFormat.format(calRepeatTimeTime.getTime());
16.     String repeatTimeMessage = edtRepeatingMessage.getText().toString();
17.
18.     alarmPreference.setRepeatingTime(repeatingTimeTime);
19.     alarmPreference.setRepeatingMessage(repeatingTimeMessage);
20.
21.     setRepeatingText();
22.
23.     alarmReceiver.setRepeatingAlarm(this, AlarmReceiver.TYPE_REPEATING,
24.         alarmPreference.getRepeatingTime(), alarmPreference.getRepeatingMessage());
25.
26. }
  
```

4. Sehingga secara keseluruhan kode method `onClick()` kita menjadi seperti ini :

```

1. @Override
2.     public void onClick(View v) {
3.         if (v.getId() == R.id.btn_one_time_alarm_date){
4.             final Calendar currentDate = Calendar.getInstance();
5.             new DatePickerDialog(this, new DatePickerDialog.OnDateSe
6. tListener() {
7.                 @Override
8.                 public void onDateSet(DatePicker view, int year, int
9. monthOfYear, int dayOfMonth) {
10.                     calOneTimeDate.set(year, monthOfYear, dayOfMonth
11. );
12.                     SimpleDateFormat dateFormat = new SimpleDateFormat
13. at("yyyy-MM-dd");
14.                     tvOneTimeDate.setText(dateFormat.format(calOneTi
15. meDate.getTime()));
16.                 }
17.             }, currentDate.get(Calendar.YEAR), currentDate.get(Calen
18. dar.MONTH), currentDate.get(Calendar.DATE)).show();
19.         }
20.         else if (v.getId() == R.id.btn_one_time_alarm_time){
21.             final Calendar currentDate = Calendar.getInstance();
22.             new TimePickerDialog(this, new TimePickerDialog.OnTimeSe
23. tListener() {
24.                 @Override
25.                 public void onTimeSet(TimePicker view, int hourOfDay
26. , int minute) {
27.                     calOneTimeTime.set(Calendar.HOUR_OF_DAY, hourOfD
28. ay);
29.                     calOneTimeTime.set(Calendar.MINUTE, minute);
30.                     SimpleDateFormat dateFormat = new SimpleDateFormat
31. at("HH:mm");
32.                     tvOneTimeTime.setText(dateFormat.format(calOneTi
33. meTime.getTime()));
34.                 }
35.             }, currentDate.get(Calendar.HOUR_OF_DAY), currentDate.ge
36. t(Calendar.MINUTE), true).show();
37.         }
38.         }else if (v.getId() == R.id.btn_repeating_time_alarm_time){
39.             final Calendar currentDate = Calendar.getInstance();
40.             new TimePickerDialog(this, new TimePickerDialog.OnTimeSe
41. tListener() {
42.                 @Override
43.                 public void onTimeSet(TimePicker view, int hourOfDay
44. , int minute) {
45.                     calRepeatTimeTime.set(Calendar.HOUR_OF_DAY, hour
46. OfDay);
47.                     calRepeatTimeTime.set(Calendar.MINUTE, minute);
48.                     SimpleDateFormat dateFormat = new SimpleDateFormat
49. at("HH:mm");
50.                     tvRepeatingTime.setText(dateFormat.format(calRep
51. eatTimeTime.getTime()));
52.                     //Log.v(TAG, "The choosen one " + date.getTime()
53. );
54.                 }
55.             }
56.         }

```



```

37.         }, currentDate.get(Calendar.HOUR_OF_DAY), currentDate.ge
t(Calendar.MINUTE), true).show();
38.     }
39.     else if (v.getId() == R.id.btn_set_one_time_alarm) {
40.
41.         SimpleDateFormat dateFormat = new SimpleDateFormat("
yyyy-MM-dd");
42.         String oneTimeDate = dateFormat.format(calOneTimeDat
e.getTime());
43.         SimpleDateFormat timeFormat = new SimpleDateFormat("
HH:mm");
44.         String oneTimeTime = timeFormat.format(calOneTimeTim
e.getTime());
45.         String oneTimeMessage = edtOneTimeMessage.getText().
toString();
46.
47.         alarmPreference.setOneTimeDate(oneTimeDate);
48.         alarmPreference.setOneTimeMessage(oneTimeMessage);
49.         alarmPreference.setOneTimeTime(oneTimeTime);
50.
51.         //setOneTimeText();
52.
53.         alarmReceiver.setOneTimeAlarm(this, AlarmReceiver.TY
PE_ONE_TIME,
54.                                     alarmPreference.getOneTimeDate(),
55.                                     alarmPreference.getOneTimeTime(),
56.                                     alarmPreference.getOneTimeMessage());
57.
58.     }
59.     else if (v.getId() == R.id.btn_repeating_time_alarm){
60.         SimpleDateFormat timeFormat = new SimpleDateFormat("HH:m
m");
61.
62.         String repeatTimeTime = timeFormat.format(calRepeatTimeT
ime.getTime());
63.         String repeatTimeMessage = edtRepeatingMessage.getText()
.toString();
64.         alarmPreference.setRepeatingTime(repeatTimeTime);
65.         alarmPreference.setRepeatingMessage(repeatTimeMessage);
66.
67.         alarmReceiver.setRepeatingAlarm(this, AlarmReceiver.TYPE
_REPEATING,
68.                                     alarmPreference.getRepeatingTime(), alarmPrefere
nce.getRepeatingMessage());
69.
70.     }
71.     else if (v.getId() == R.id.btn_cancel_repeating_alarm){
72.         alarmReceiver.cancelAlarm(this, AlarmReceiver.TYPE_REPEA
TING);
73.     }
74. }

```

5. Kemudian lengkapi dengan membuat method `setRepeatingText()` seperti berikut :

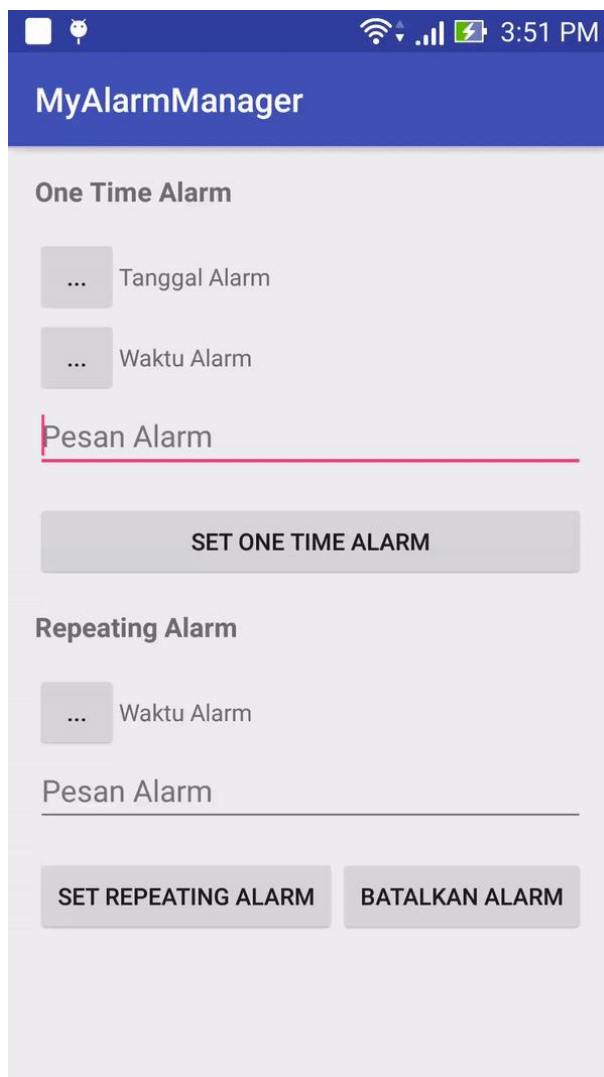
```
1. private void setRepeatingText() {
```

```
2.     tvRepeatingTime.setText(alarmPreference.getRepeatingTime());
3.     edtRepeatingMessage.setText(alarmPreference.getRepeatingMessage(
4.     ));
```

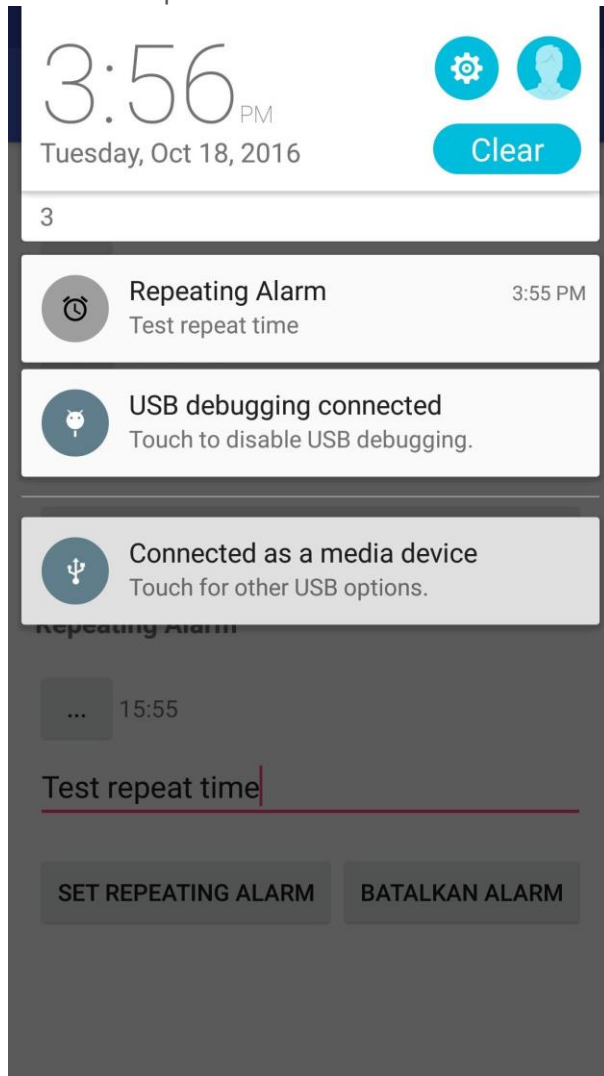
6. Silakan tambahkan baris berikut di method `onCreate()` yang dapat memberikan nilai default yang pernah kita gunakan ketika aplikasi kita buka

```
1. if (!TextUtils.isEmpty(alarmPreference.getRepeatingTime())){
2.     setRepeatingText();
3. }
```

7. Setelah selesai, silakan jalankan aplikasi dan pilih waktu yang diinginkan untuk menampilkan notifikasi harian seperti berikut.



Jika berhasil seharusnya notifikasi akan muncul setiap hari pada waktu yang kamu tentukan seperti ini.



Keren! Saat ini anda sudah bisa membuat sebuah repeating-alarm yang kedepannya bisa digunakan untuk menjadwalkan proses (*scheduling task*). Misalnya melakukan sinkronisasi ke server. Thumbs Up!

Sebelumnya, kami belum menjelaskan bagaimana membatalkan sebuah alarm manager yang sudah dipasang. Modul ini akan mencoba membedah mekanisme tersebut.

1. Berkas `activity_main.xml` sekarang seharusnya berisi kode berikut :

```
1. <Button
2.     android:id="@+id/btn_repeating_time_alarm"
3.     android:layout_width="match_parent"
4.     android:layout_height="wrap_content"
5.     android:text="Set Repeating Alarm"/>
```

Ubah kode di atas sehingga menjadi :

```
6. <LinearLayout
7.     android:layout_width="match_parent"
8.     android:layout_height="wrap_content"
9.     android:orientation="horizontal">
10.    <Button
11.        android:id="@+id/btn_repeating_time_alarm"
12.        android:layout_width="match_parent"
13.        android:layout_height="wrap_content"
14.        android:text="Set Repeating Alarm"
15.        android:layout_weight="0.5"/>
16.    <Button
17.        android:id="@+id/btn_cancel_repeating_alarm"
18.        android:layout_width="match_parent"
19.        android:layout_height="wrap_content"
20.        android:text="Batalan Alarm"
21.        android:layout_weight="0.5"/>
22.</LinearLayout>
```

2. Selanjutnya pada `MainActivity` tambahkan `btnCancelRepeating` pada deklarasi obyek `Button` seperti ini :

```
1. private Button btnOneTimeDate, btnOneTimeTime, btnOneTime ,btnRepeat
   ingTime ,btnRepeating, btnCancelRepeatingAlarm;
```

Seperti biasa, lakukan casting obyek view `btnCancelRepeatingAlarm` agar view tersebut dapat kita manipulasi seperti ini :

```
2. btnCancelRepeatingAlarm = (Button)findViewById(R.id.btn_cancel_repea
   ting_alarm);
3. btnCancelRepeatingAlarm.setOnClickListener(this);
```

3. Selanjutnya kita tambahkan kode seleksi baru pada method `onClick()` yang akan menjalankan proses pembatalan repeating alarm ketika tombol **Batalan Alarm** diklik.

```
1. else if (v.getId() == R.id.btn_cancel_repeating_alarm){
2.     alarmReceiver.cancelAlarm(this, AlarmReceiver.TYPE_REPEATING);
3. }
```

Sehingga kode `MainActivity` kita secara keseluruhan menjadi seperti berikut.

```
4. public class MainActivity extends AppCompatActivity implements View.
   OnClickListener {
5.     private TextView tvOneTimeDate, tvOneTimeTime;
6.     private TextView tvRepeatingTime;
7.     private EditText edtOneTimeMessage, edtRepeatingMessage;
8.     private Button btnOneTimeDate, btnOneTimeTime, btnOneTime, btnRe
       peatingTime, btnRepeating, btnCancelRepeatingAlarm;
9.
10.    private Calendar calOneTimeDate, calOneTimeTime, calRepeatTimeTim
       e;
11.
12.    private AlarmReceiver alarmReceiver;
13.    private AlarmPreference alarmPreference;
14.
15.    @Override
16.    protected void onCreate(Bundle savedInstanceState) {
17.        super.onCreate(savedInstanceState);
18.        setContentView(R.layout.activity_main);
19.        setTitle("MyAlarmManager");
20.        tvOneTimeDate = (TextView)findViewById(R.id.tv_one_time_alar
       m_date);
21.        tvOneTimeTime = (TextView)findViewById(R.id.tv_one_time_alar
       m_time);
22.        edtOneTimeMessage = (EditText)findViewById(R.id.edt_one_time
       _alarm_message);
23.        btnOneTimeDate = (Button)findViewById(R.id.btn_one_time_alar
       m_date);
24.        btnOneTimeTime = (Button)findViewById(R.id.btn_one_time_alar
       m_time);
25.        btnOneTime = (Button)findViewById(R.id.btn_set_one_time_alar
       m);
26.
27.        tvRepeatingTime = (TextView)findViewById(R.id.tv_repeating_a
       larm_time);
28.        edtRepeatingMessage = (EditText)findViewById(R.id.edt_repea
       ting_alarm_message);
29.        btnRepeatingTime = (Button)findViewById(R.id.btn_repeating_t
       ime_alarm_time);
30.        btnRepeating = (Button)findViewById(R.id.btn_repeating_time_
       alarm);
31.
32.        btnCancelRepeatingAlarm = (Button)findViewById(R.id.btn_canc
       el_repeating_alarm);
33.
34.        btnOneTimeDate.setOnClickListener(this);
35.        btnOneTimeTime.setOnClickListener(this);
```

```

36.         btnOneTime.setOnClickListener(this);
37.         btnRepeatingTime.setOnClickListener(this);
38.         btnRepeating.setOnClickListener(this);
39.         btnCancelRepeatingAlarm.setOnClickListener(this);
40.
41.         calOneTimeDate = Calendar.getInstance();
42.         calOneTimeTime = Calendar.getInstance();
43.         calRepeatTimeTime = Calendar.getInstance();
44.
45.         alarmPreference = new AlarmPreference(this);
46.         alarmReceiver = new AlarmReceiver();
47.
48.         if (!TextUtils.isEmpty(alarmPreference.getOneTimeDate())){
49.             setOneTimeText();
50.         }
51.
52.         if (!TextUtils.isEmpty(alarmPreference.getRepeatingTime())){
53.             setRepeatingText();
54.         }
55.
56.     }
57.
58.     @Override
59.     public void onClick(View v) {
60.         if (v.getId() == R.id.btn_one_time_alarm_date){
61.             final Calendar currentDate = Calendar.getInstance();
62.             new DatePickerDialog(this, new DatePickerDialog.OnDateSe
tListener() {
63.                 @Override
64.                 public void onDateSet(DatePicker view, int year, int
monthOfYear, int dayOfMonth) {
65.                     calOneTimeDate.set(year, monthOfYear, dayOfMon
th);
66.                     SimpleDateFormat dateFormat = new SimpleDateForm
at("yyyy-MM-dd");
67.                     tvOneTimeDate.setText(dateFormat.format(calOneTi
meDate.getTime()));
68.                 }
69.             }, currentDate.get(Calendar.YEAR), currentDate.get(Calen
dar.MONTH), currentDate.get(Calendar.DATE)).show();
70.         }
71.         else if (v.getId() == R.id.btn_one_time_alarm_time){
72.             final Calendar currentDate = Calendar.getInstance();
73.             new TimePickerDialog(this, new TimePickerDialog.OnTimeSe
tListener() {
74.                 @Override
75.                 public void onTimeSet(TimePicker view, int hourOfDay
, int minute) {
76.                     calOneTimeTime.set(Calendar.HOUR_OF_DAY, hourOfD
ay);
77.                     calOneTimeTime.set(Calendar.MINUTE, minute);
78.                     SimpleDateFormat dateFormat = new SimpleDateForm
at("HH:mm");
79.                     tvOneTimeTime.setText(dateFormat.format(calOneTi
meTime.getTime()));
80.                 }
81.             }, currentDate.get(Calendar.HOUR_OF_DAY), currentDate.ge
t(Calendar.MINUTE), true).show();
82.         }

```

```

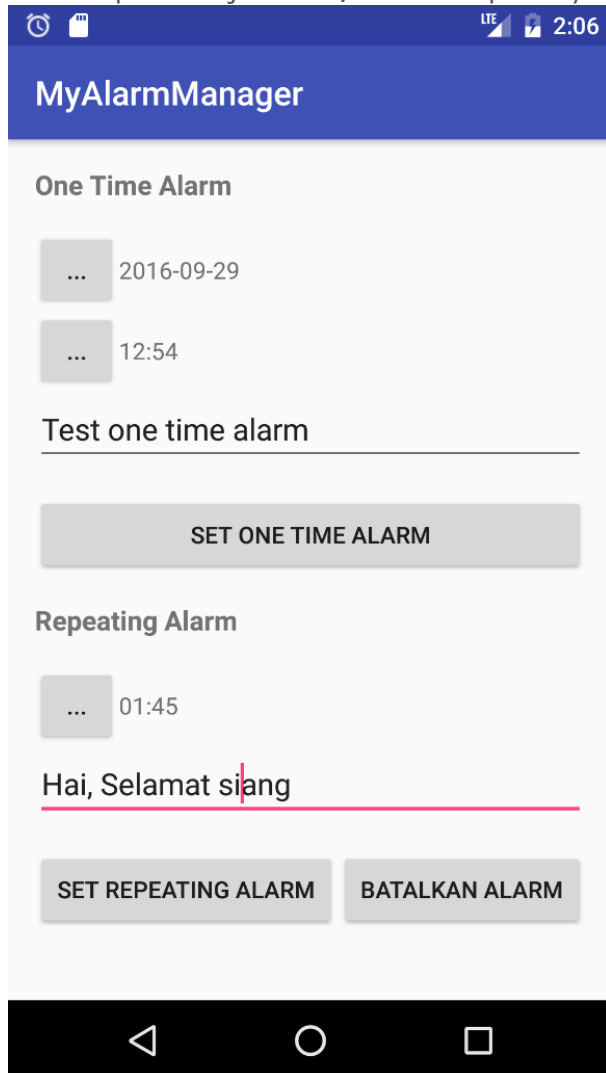
83.         }else if (v.getId() == R.id.btn_repeating_time_alarm_time){
84.             final Calendar currentDate = Calendar.getInstance();
85.             new TimePickerDialog(this, new TimePickerDialog.OnTimeSe
            tListener() {
86.                 @Override
87.                 public void onTimeSet(TimePicker view, int hourOfDay
            , int minute) {
88.                     calRepeatTimeTime.set(Calendar.HOUR_OF_DAY, hour
            OfDay);
89.                     calRepeatTimeTime.set(Calendar.MINUTE, minute);
90.                     SimpleDateFormat dateFormat = new SimpleDateForm
            at("HH:mm");
91.                     tvRepeatingTime.setText(dateFormat.format(calRep
            eatTimeTime.getTime()));
92.
93.                 }
94.             }, currentDate.get(Calendar.HOUR_OF_DAY), currentDate.ge
            t(Calendar.MINUTE), true).show();
95.         }
96.         else if (v.getId() == R.id.btn_set_one_time_alarm) {
97.
98.             SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy
            -MM-dd");
99.
100.            String oneTimeDate = dateFormat.format(calOneTimeDate
            .getTime());
101.
102.            SimpleDateFormat timeFormat = new SimpleDateFormat("H
            H:mm");
103.
104.            String oneTimeTime = timeFormat.format(calOneTimeTime
            .getTime());
105.            String oneTimeMessage = edtOneTimeMessage.getText().t
            oString();
106.
107.            alarmPreference.setOneTimeDate(oneTimeDate);
108.            alarmPreference.setOneTimeMessage(oneTimeMessage);
109.            alarmPreference.setOneTimeTime(oneTimeTime);
110.
111.            alarmReceiver.setOneTimeAlarm(this, AlarmReceiver.TYP
            E_ONE_TIME,
112.                alarmPreference.getOneTimeDate(),
113.                alarmPreference.getOneTimeTime(),
114.                alarmPreference.getOneTimeMessage());
115.
116.        }
117.        else if (v.getId() == R.id.btn_repeating_time_alarm){
118.            SimpleDateFormat timeFormat = new SimpleDateFormat("H
            H:mm");
119.
120.            String repeatTimeTime = timeFormat.format(calRepeatTi
            meTime.getTime());
121.            String repeatTimeMessage = edtRepeatingMessage.getTex
            t().toString();
122.            alarmPreference.setRepeatingTime(repeatTimeTime);
123.            alarmPreference.setRepeatingMessage(repeatTimeMessage
            );
124.
125.

```



```
126.         alarmReceiver.setRepeatingAlarm(this, AlarmReceiver.T
    TYPE_REPEATING,
127.         alarmPreference.getRepeatingTime(), alarmPref
    erence.getRepeatingMessage());
128.
129.     }
130.     else if (v.getId() == R.id.btn_cancel_repeating_alarm){
131.         alarmReceiver.cancelAlarm(this, AlarmReceiver.TYPE_RE
    PEATING);
132.     }
133.
134. }
135.
136. private void setOneTimeText() {
137.     tvOneTimeTime.setText(alarmPreference.getOneTimeTime());
138.     tvOneTimeDate.setText(alarmPreference.getOneTimeDate());
139.     edtOneTimeMessage.setText(alarmPreference.getOneTimeMessa
    ge());
140. }
141.
142. private void setRepeatingText() {
143.     tvRepeatingTime.setText(alarmPreference.getRepeatingTime(
    ));
144.     edtRepeatingMessage.setText(alarmPreference.getRepeatingM
    essage());
145. }
146. }
```

4. Ketika aplikasi dijalankan, maka tampilannya kurang lebih menjadi seperti ini.



5. Terakhir, tambahkan sebuah **Toast** seperti berikut :

```
1. Toast.makeText(context, "Repeating alarm dibatalkan", Toast.LENGTH_SHORT).show();
```

6. Metode cancelAlarm() akan berubah menjadi :

```
1. public void cancelAlarm(Context context, String type){
2.     AlarmManager alarmManager = (AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
3.     Intent intent = new Intent(context, AlarmReceiver.class);
4.     int requestCode = type.equalsIgnoreCase(TYPE_ONE_TIME) ? NOTIF_ID_ONETIME : NOTIF_ID_REPEATING;
5.     PendingIntent pendingIntent = PendingIntent.getBroadcast(context, requestCode, intent, 0);
6.     alarmManager.cancel(pendingIntent);
7.     Toast.makeText(context, "Repeating alarm dibatalkan", Toast.LENGTH_SHORT).show();
8. }
```

Selesai! Sekarang mari kita uji kodenya. Skenarionya seperti berikut :

1. Ubah nilai pada variabel `repeatingTime` di `MainActivity` sesuai dengan waktu yang mendekati saat ini (bedakan sekitar 5 menit dari sekarang). Sebagai contoh, saat ini waktu dipasang pada pukul 01.45
2. Klik tombol **SET REPEATING ALARM** untuk menjalankan alarm untuk waktu yang ditentukan
3. Klik tombol **BATALKAN ALARM** untuk membatalkan alarm yang baru saja dijalankan.
4. Jika sukses seharusnya pada waktu yang diberikan tidak tampil notifikasi alarm seperti sebelumnya.

Lembar Kerja Praktikum

NPM:	Nama Asisten:
Nama:	Nilai:
Kelas:	

Form ini digunakan untuk menilai kerja mahasiswa selama proses praktikum berlangsung. Selama praktikum mahasiswa diminta untuk membuat Aplikasi Alarm Manager. Semua tombol harus dapat berfungsi sesuai dengan materi yang telah disampaikan.

