[Scheduler](https://youtu.be/mXK4kSt4rLo)



Teori :

Adakalanya kita membutuhkan perlu suatu mekanisme untuk mengulang sebuah proses di waktu tertentu. Misalnya, kita ingin memiliki fungsi pengingat (*reminder*) pada aplikasi *event*, jadwal tayang film di bioskop, atau aplikasi alarm untuk mengingatkan kita untuk bangun di pagi hari. Disinilah alarm manager berperan penting.

Alarm manager akan memberikan kemudahan bagi aplikasi untuk melakukan operasi berbasis waktu diluar daur hidup aplikasi itu sendiri.

Contohnya seperti yang baru saja disebutkan diatas. Umumnya alarm memiliki beberapa karakteristik, antara lain :

1. Dapat menjalankan obyek intent berdasarkan waktu dan interval yang ditentukan.
2. Bisa bekerja dengan baik dengan broadcast receiver untuk menjalankan komponen lain seperti service untuk melakukan operasi tertentu.
3. Alarm berjalan diluar daur hidup aplikasi induknya sehingga kita bisa merancang fungsi alarm untuk melakukan sebuah aksi ketika aplikasi sedang tidak dijalankan bahkan ketika peranti dalam keadaan *idle* atau *sleep.*
4. Esensi utama dari alarm manager adalah untuk meminimalkan penggunaan *resource* dan menghindari penggunaan *timer* dan background service yang berkepanjangan untuk melakukan operasi atau aksi yang dibutuhkan.

Penggunaan dan perancangan alarm manager yang tepat akan membuat aplikasi kita menjadi lebih baik, serta tentunya memberikan pengalaman yang juga berkesan. Namun sebaliknya desain yang buruk akan menjadi bumerang bagi aplikasi kita. Akibat dari rancangan yang tidak sesuai akan menguras baterai di peranti kita.

Penggunaan alarm manager yang baik adalah dengan menerapkan cara berikut ini:

1. Lakukan hanya untuk proses di lokal
2. Jaga frekuensi alarm yang dijalankan dengan interval yang tidak pendek
3. Jika tidak terlalu penting, hindari penggunaan tipe alarm yang dapat membangunkan peranti seperti RTC\_WAKEUP
4. Jika alarm manager digunakan untuk menjadwalkan *scheduling task* yang terhubung ke jaringan dan melakukan *request* ke *server*, misalnya unduh data terbaru atau sinkronisasi data, maka hindari penggunaan setRepeating() dan gunakan setInExactRepeating() dengan penggunaan tipe alarm berbasis ELAPSED\_REALTIME. Pendekatan tersebut merupakan strategi untuk mengelompokkan proses alarm dari aplikasi yang berbeda dengan dijalankan secara bersamaan agar lebih menghemat daya baterai.

**Catatan : Di atas API 19 KitKat semua pemanggilan *repeating alarm*adalah menggunakan *in exact repeating.*Baca tautan berikut**[**ini**](https://developer.android.com/reference/android/app/AlarmManager.html#setInexactRepeating(int,%20long,%20long,%20android.app.PendingIntent))**.**

Untuk lebih mendalami topik alarm manager, Anda dapat membaca tautan berikut ini:

* [Alarm Manager](https://developer.android.com/reference/android/app/AlarmManager.html)
* [Scheduling Alarm](https://developer.android.com/training/scheduling/alarms.html#/h)

*From <*[*https://www.dicoding.com/academies/14/tutorials/187?from=139*](https://www.dicoding.com/academies/14/tutorials/187?from=139)*>*

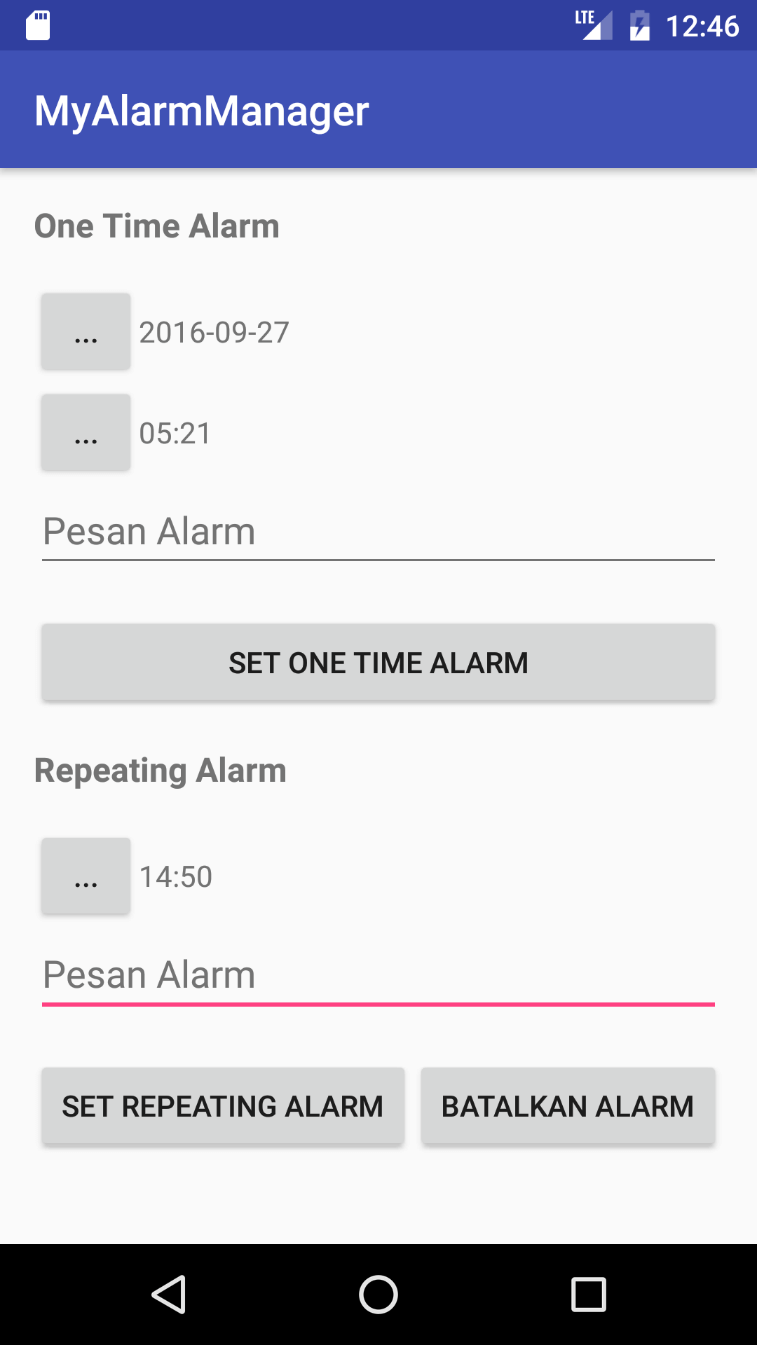
Tujuan

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

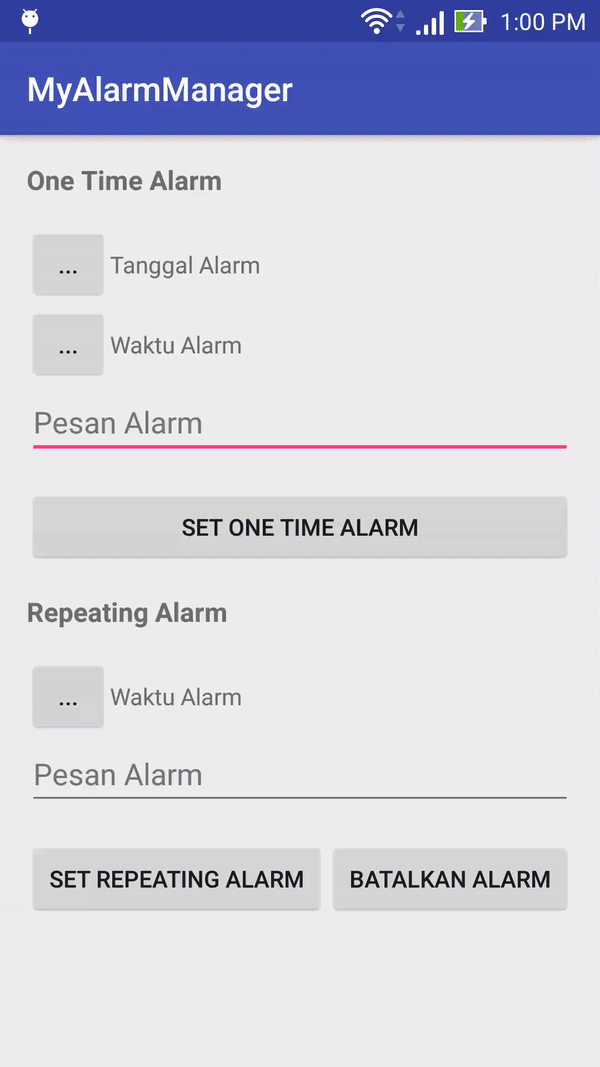
* 1. Aplikasi akan memiliki kemampuan untuk melakukan *one-time-execution alarm* dan alarm yang berulang berdasarkan waktu yang ditentukan.
  2. Aplikasi akan menyimpan beberapa nilai untuk kedua tipe alarm tersebut.
  3. Aplikasi akan menampilkan notifikasi dengan suara dan getar ketika alarm dijalankan.

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\_margin"/>**
     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. **<TextView**
     46. **android:id="@+id/tv\_one\_time\_alarm\_time"**
     47. **android:layout\_width="wrap\_content"**
     48. **android:layout\_height="wrap\_content"**
     49. **android:text="Waktu Alarm"**
     50. **android:layout\_marginBottom="@dimen/activity\_vertical\_margin"/>**
     51. **</LinearLayout>**
     52. **<EditText**
     53. **android:id="@+id/edt\_one\_time\_alarm\_message"**
     54. **android:layout\_width="match\_parent"**
     55. **android:layout\_height="wrap\_content"**
     56. **android:hint="Pesan Alarm"**
     57. **android:layout\_marginBottom="@dimen/activity\_vertical\_margin"/>**
     58. **<Button**
     59. **android:id="@+id/btn\_set\_one\_time\_alarm"**
     60. **android:layout\_width="match\_parent"**
     61. **android:layout\_height="wrap\_content"**
     62. **android:text="Set One Time Alarm"**
     63. **android:layout\_marginBottom="@dimen/activity\_vertical\_margin"/>**
     64. **<TextView**
     65. **android:layout\_width="match\_parent"**
     66. **android:layout\_height="wrap\_content"**
     67. **android:text="Repeating Alarm"**
     68. **android:layout\_marginBottom="@dimen/activity\_vertical\_margin"**
     69. **android:textSize="16sp"**
     70. **android:textStyle="bold"/>**
     71. **<LinearLayout**
     72. **android:layout\_width="match\_parent"**
     73. **android:layout\_height="wrap\_content"**
     74. **android:orientation="horizontal">**
     75. **<Button**
     76. **android:text="..."**
     77. **android:layout\_width="50dp"**
     78. **android:layout\_height="wrap\_content"**
     79. **android:id="@+id/btn\_repeating\_time\_alarm\_time" />**
     80. **<TextView**
     81. **android:id="@+id/tv\_repeating\_alarm\_time"**
     82. **android:layout\_width="wrap\_content"**
     83. **android:layout\_height="wrap\_content"**
     84. **android:text="Waktu Alarm"**
     85. **android:layout\_marginBottom="@dimen/activity\_vertical\_margin"/>**
     86. **</LinearLayout>**
     87. **<EditText**
     88. **android:id="@+id/tv\_repeating\_alarm\_message"**
     89. **android:layout\_width="match\_parent"**
     90. **android:layout\_height="wrap\_content"**
     91. **android:hint="Pesan Alarm"**
     92. **android:layout\_marginBottom="@dimen/activity\_vertical\_margin"/>**
     93. **<LinearLayout**
     94. **android:layout\_width="match\_parent"**
     95. **android:layout\_height="wrap\_content"**
     96. **android:orientation="horizontal">**
     97. **<Button**
     98. **android:id="@+id/btn\_repeating\_time\_alarm"**
     99. **android:layout\_width="wrap\_content"**
     100. **android:layout\_height="wrap\_content"**
     101. **android:text="Set Repeating Alarm"**
     102. **android:layout\_weight="0.5"**
     103. **/>**
     104. **<Button**
     105. **android:id="@+id/btn\_cancel\_repeating\_alarm"**
     106. **android:layout\_width="wrap\_content"**
     107. **android:layout\_height="wrap\_content"**
     108. **android:text="Batalkan Alarm"**
     109. **android:layout\_weight="0.5"**
     110. **/>**
     111. **</LinearLayout>**
     112. **</LinearLayout>**Sehingga tampilan aplikasi dari activity\_main.xml ketika dijalankan seperti berikut :



* 1. 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, Context.*MODE\_PRIVATE*);
     12. editor = mSharedPreferences.edit();
     13. }
     14. public void setOneTimeDate(String date){
     15. editor.putString(KEY\_ONE\_TIME\_DATE, date);
     16. editor.commit();
     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, null);
     34. }
     35. public void setRepeatingTime(String time){
     36. editor.putString(KEY\_REPEATING\_TIME, time);
     37. editor.commit();
     38. }
     39. public String getRepeatingTime(){
     40. return mSharedPreferences.getString(KEY\_REPEATING\_TIME, null);
     41. }
     42. public void setRepeatingMessage(String message){
     43. editor.putString(KEY\_REPEATING\_MESSAGE, message);
     44. editor.commit();
     45. }
     46. public String getRepeatingMessage(){
     47. return mSharedPreferences.getString(KEY\_REPEATING\_MESSAGE, null);
     48. }
     49. public void clear(){
     50. editor.clear();
     51. editor.commit();
     52. }
     53. }
  2. 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;**
     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*(context, 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](https://design.google.com/icons/), dibagian kategori peranti. Setelah diunduh, masukkan ikon ke dalam folder **res → drawable**.
  3. 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 ,btnRepeatingTime ,btnRepeating,  btnCancelRepeatingAlarm;**
     6. **private Calendar calOneTimeDate, calOneTimeTime , calRepeatTimeTime;**
     7. **private AlarmReceiver alarmReceiver;**
     8. **private AlarmPreference alarmPreference;**
     10. @Override
     11. protected void onCreate(Bundle savedInstanceState) {
     12. super.onCreate(savedInstanceState);
     13. setContentView(R.layout.activity\_main);
     15. **tvOneTimeDate = (TextView)findViewById(R.id.tv\_one\_time\_alarm\_date);**
     16. **tvOneTimeTime = (TextView)findViewById(R.id.tv\_one\_time\_alarm\_time);**
     17. **edtOneTimeMessage = (EditText)findViewById(R.id.edt\_one\_time\_alarm\_message);**
     18. **btnOneTimeDate = (Button)findViewById(R.id.btn\_one\_time\_alarm\_date);**
     19. **btnOneTimeTime = (Button)findViewById(R.id.btn\_one\_time\_alarm\_time);**
     20. **btnOneTime = (Button)findViewById(R.id.btn\_set\_one\_time\_alarm);**
     21. **tvRepeatingTime = (TextView)findViewById(R.id.tv\_repeating\_alarm\_time);**
     22. **edtRepeatingMessage = (EditText)findViewById(R.id.edt\_repeating\_alarm\_message);**
     23. **btnRepeatingTime = (Button)findViewById(R.id.btn\_repeating\_time\_alarm\_time);**
     24. **btnRepeating = (Button)findViewById(R.id.btn\_repeating\_time\_alarm);**
     25. **btnCancelRepeatingAlarm = (Button)findViewById(R.id.btn\_cancel\_repeating\_alarm);**
     26. **btnOneTimeDate.setOnClickListener(this);**
     27. **btnOneTimeTime.setOnClickListener(this);**
     28. **btnOneTime.setOnClickListener(this);**
     29. **btnRepeatingTime.setOnClickListener(this);**
     30. **btnRepeating.setOnClickListener(this);**
     31. **btnCancelRepeatingAlarm.setOnClickListener(this);**
     32. **calOneTimeDate = Calendar.getInstance();**
     33. **calOneTimeTime = Calendar.getInstance();**
     34. **calRepeatTimeTime = Calendar.*getInstance*();**
     35. **alarmPreference = new AlarmPreference(this);**
     36. **alarmReceiver = new AlarmReceiver();**
     37. **if (!TextUtils.isEmpty(alarmPreference.getOneTimeDate())){**
     38. **setOneTimeText();**
     39. **}**
     41. }
     42. **@Override**
     43. **public void onClick(View v) {**
     44. **if (v.getId() == R.id.btn\_one\_time\_alarm\_date){**
     45. **final Calendar currentDate = Calendar.getInstance();**
     46. **new DatePickerDialog(this, new DatePickerDialog.OnDateSetListener() {**
     47. **@Override**
     48. **public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {**
     49. **calOneTimeDate.set(year, monthOfYear, dayOfMonth);**
     50. **SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");**
     51. **tvOneTimeDate.setText(dateFormat.format(calOneTimeDate.getTime()));**
     52. **}**
     53. **}, currentDate.get(Calendar.YEAR), currentDate.get(Calendar.MONTH), currentDate.get(Calendar.DATE)).show();**
     54. **}**
     55. **else if (v.getId() == R.id.btn\_one\_time\_alarm\_time){**
     56. **final Calendar currentDate = Calendar.getInstance();**
     57. **new TimePickerDialog(this, new TimePickerDialog.OnTimeSetListener() {**
     58. **@Override**
     59. **public void onTimeSet(TimePicker view, int hourOfDay, int minute) {**
     60. **calOneTimeTime.set(Calendar.HOUR\_OF\_DAY, hourOfDay);**
     61. **calOneTimeTime.set(Calendar.MINUTE, minute);**
     62. **SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm");**
     63. **tvOneTimeTime.setText(dateFormat.format(calOneTimeTime.getTime()));**
     64. ***//Log.v(TAG, "The choosen one " + date.getTime());***
     65. **}**
     66. **}, currentDate.get(Calendar.HOUR\_OF\_DAY), currentDate.get(Calendar.MINUTE), true).show();**
     67. **}**
     68. **else if (v.getId() == R.id.btn\_set\_one\_time\_alarm){**
     69. **SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");**
     70. **String oneTimeDate = dateFormat.format(calOneTimeDate.getTime());**
     71. **SimpleDateFormat timeFormat = new SimpleDateFormat("HH:mm");**
     72. **String oneTimeTime = timeFormat.format(calOneTimeTime.getTime());**
     73. **String oneTimeMessage = edtOneTimeMessage.getText().toString();**
     74. **alarmPreference.setOneTimeDate(oneTimeDate);**
     75. **alarmPreference.setOneTimeMessage(oneTimeMessage);**
     76. **alarmPreference.setOneTimeTime(oneTimeTime);**
     77. **alarmReceiver.setOneTimeAlarm(this, AlarmReceiver.TYPE\_ONE\_TIME,**
     78. **alarmPreference.getOneTimeDate(),**
     79. **alarmPreference.getOneTimeTime(),**
     80. **alarmPreference.getOneTimeMessage());**
     81. **}**
     82. **}**
     84. **private void setOneTimeText() {**
     85. **tvOneTimeTime.setText(alarmPreference.getOneTimeTime());**
     86. **tvOneTimeDate.setText(alarmPreference.getOneTimeDate());**
     87. **edtOneTimeMessage.setText(alarmPreference.getOneTimeMessage());**
     88. **}**
     89. }
  4. 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.
  5. 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.

**Bedah Kode**

Kita mulai dengan skenario nya terlebih dahulu.

Kode pada aplikasi diatas berfungsi untuk menjalankan sebuah alarm yang bersifat *one-time-execution*. Logika dasarnya adalah, kita memanfaatkan fasilitas alarm manager yang terdapat didalam framework Android untuk menjalankan tugas atau proses tertentu berdasarkan waktu yang ditentukan.

Di sini kita akan melakukan proses penampilan notifikasi berdasarkan nilai yang dimasukkan oleh pengguna. Baik itu tanggal, waktu dan pesannya.

Ketika waktu yang ditentukan cocok dengan waktu yang berjalan pada sistem, maka metode onReceive pada BroadcastReceiverakan dijalankan.

Logika yang sederhana bukan?

Ada beberapa istilah baru yang bisa jadi teman-teman baru menemukannya pada modul ini. Untuk memahaminya, mari kita lanjut pembedahan kodenya.

Kita awali dengan pembuatan kelas AlarmPreference. Idenya adalah kita dapat menyimpan nilai-nilai terakhir yang kita set sewaktu membuat obyek alarm. Nilai-nilai seperti tanggal, jam, dan pesan akan disimpan ke dalam sebuah mekanisme penyimpanan berbasis pasangan kunci dan nilai (*key value pair*). Mekanisme tersebut memanfaatkan SharedPreference. Semua nilai kita simpan ke dalam tipe data string seperti ini.

* 1. alarmPreference.setOneTimeDate(oneTimeDate);
  2. alarmPreference.setOneTimeMessage(oneTimeMessage);
  3. alarmPreference.setOneTimeTime(oneTimeTime);

Ketika pengguna membuka kembali aplikasi, nilai-nilai di atas akan ditampilkan ke dalam obyek TextView.

* 1. private void setOneTimeText() {
  2. tvOneTimeTime.setText(alarmPreference.getOneTimeTime());
  3. tvOneTimeDate.setText(alarmPreference.getOneTimeDate());
  4. edtOneTimeMessage.setText(alarmPreference.getOneTimeMessage());
  5. }

Anda mempelajari sesuatu yang beda saat ini. Sebuah mekanisme penyimpanan yang paling sederhana dan paling mudah digunakan.

Esensi dari kelas AlarmPreference adalah untuk menyimpan nilai terakhir yang dimasukkan pengguna sewaktu memasang alarm. Anda dapat menyimpan beragam tipe data selain string. Misalnya data bertipekan double, int, boolean, float, long dan stringset.

Anda akan mendapatkan detail tentang SharedPreference lebih mendalam lagi pada modul ke 3 nanti. 

Selanjutnya pada metode onclick, terdapat beberapa kondisi if-else yang berguna untuk membedakan button mana yang disentuh. Button dengan id btn\_one\_time\_alarm\_datedan btn\_one\_time\_alarm\_time memiliki fungsi menampilkan dialog yang menampilkan date picker dan time picker.

* 1. if (v.getId() == R.id.*btn\_one\_time\_alarm\_date*){
  2. final Calendar currentDate = Calendar.*getInstance*();
  3. new DatePickerDialog(this, new DatePickerDialog.OnDateSetListener() {
  4. @Override
  5. public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {
  6. calOneTimeDate.set(year, monthOfYear, dayOfMonth);
  7. SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
  8. tvOneTimeDate.setText(dateFormat.format(calOneTimeDate.getTime()));
  9. }
  10. }, currentDate.get(Calendar.*YEAR*), currentDate.get(Calendar.*MONTH*), currentDate.get(Calendar.*DATE*)).show();
  11. }
  12. else if (v.getId() == R.id.*btn\_one\_time\_alarm\_time*){
  13. final Calendar currentDate = Calendar.*getInstance*();
  14. new TimePickerDialog(this, new TimePickerDialog.OnTimeSetListener() {
  15. @Override
  16. public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
  17. calOneTimeTime.set(Calendar.*HOUR\_OF\_DAY*, hourOfDay);
  18. calOneTimeTime.set(Calendar.*MINUTE*, minute);
  19. SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm");
  20. tvOneTimeTime.setText(dateFormat.format(calOneTimeTime.getTime()));
  21. *//Log.v(TAG, "The choosen one " + date.getTime());*
  22. }
  23. }, currentDate.get(Calendar.*HOUR\_OF\_DAY*), currentDate.get(Calendar.*MINUTE*), true).show();
  24. }

Selanjutnya, perhatikan pada baris berikut ini.

* 1. public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {
  2. calOneTimeDate.set(year, monthOfYear, dayOfMonth);
  3. SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
  4. tvOneTimeDate.setText(dateFormat.format(calOneTimeDate.getTime()));
  5. }

Fungsi onDateSet akan dipanggil ketika kita memilih tanggal yangkita inginkan. Kemudian kita pasang variable calOneTimeDatadengan tanggalyang kitapilih. Variable tersebut dapat digunakan untuk mengisi TextView tanggalalarm.

Untuk mendapatkan string dari variable calOneTimeDate, kitamenggunakan komponen SimpleDateFormat. Perhatikan kode “yyyy-MM-dd”, ini adalah formatuntuk menampilkan tanggal dengan format seperti 2016-09-29.

Terdapat beberapa formatyang dapatkita gunakan dengan komponen SimpleDateFormat. Untuk mengetahui lebih dalam mengenai SimpleDateformat, Anda dapat langsung menuju tautan berikut :

* + [SimpleDateFormat](https://developer.android.com/reference/java/text/SimpleDateFormat.html)

Selanjutnya, pada AlarmReceiver kita membuat beberapa variabel konstanta penanda (*flag*) yang akan digunakan di kesuluruhan bagian kode, diantaranya :

* + public static final String *TYPE\_ONE\_TIME*= "OneTimeAlarm";
  + public static final String *TYPE\_REPEATING*= "RepeatingAlarm";

Dua baris diatas adalah konstanta untuk menentukan tipe alarm. Dan selanjutnya, dua baris dibawah ini adalah konstanta untuk intent key.

* + public static final String *EXTRA\_MESSAGE*= "message";
  + public static final String *EXTRA\_TYPE*= "type";

Di sini kita menggunakan dua konstanta bertipe data integer untuk menentukan notif ID sebagai ID untuk menampilkan notifikasi kepada pengguna.

* + private final int NOTIF\_ID\_ONETIME = 100;
  + private final int NOTIF\_ID\_REPEATING = 101;

Selanjutnya, bagian penting pada metode setOneTimeAlarm() adalah :

* + public void setOneTimeAlarm(Context context, String type, String date, String time, String message){
  + AlarmManager alarmManager = (AlarmManager)context.getSystemService(Context.*ALARM\_SERVICE*);
  + Intent intent = new Intent(context, AlarmReceiver.class);
  + intent.putExtra(*EXTRA\_MESSAGE*, message);
  + intent.putExtra(*EXTRA\_TYPE*, type);
  + String dateArray[] = date.split("-");
  + String timeArray[] = time.split(":");
  + Calendar calendar = Calendar.*getInstance*();
  + calendar.set(Calendar.*YEAR*, Integer.*parseInt*(dateArray[0]));
  + calendar.set(Calendar.*MONTH*, Integer.*parseInt*(dateArray[1])-1);
  + calendar.set(Calendar.*DAY\_OF\_MONTH*, Integer.*parseInt*(dateArray[2]));
  + calendar.set(Calendar.*HOUR\_OF\_DAY*, Integer.*parseInt*(timeArray[0]));
  + calendar.set(Calendar.*MINUTE*, Integer.*parseInt*(timeArray[1]));
  + calendar.set(Calendar.*SECOND*, 0);
  + int requestCode = NOTIF\_ID\_ONETIME;
  + PendingIntent pendingIntent =  PendingIntent.*getBroadcast*(context, requestCode, intent, 0);
  + alarmManager.set(AlarmManager.*RTC\_WAKEUP*, calendar.getTimeInMillis(), pendingIntent);
  + Toast.*makeText*(context, "One time alarm set up", Toast.*LENGTH\_SHORT*).show();
  1. }

Pada kode di atas, kita membuat sebuah obyek dari kelas AlarmManager. Kita menyiapkan sebuah Intent yang akan menjalankan AlarmReceiver dan membawa data alarm dan pesan.

* 1. String dateArray[] = date.split("-");
  2. String timeArray[] = time.split(":");
  3. Calendar calendar = Calendar.*getInstance*();
  4. calendar.set(Calendar.*YEAR*, Integer.*parseInt*(dateArray[0]));
  5. calendar.set(Calendar.*MONTH*, Integer.*parseInt*(dateArray[1])-1);
  6. calendar.set(Calendar.*DAY\_OF\_MONTH*, Integer.*parseInt*(dateArray[2]));
  7. calendar.set(Calendar.*HOUR\_OF\_DAY*, Integer.*parseInt*(timeArray[0]));
  8. calendar.set(Calendar.*MINUTE*, Integer.*parseInt*(timeArray[1]));
  9. calendar.set(Calendar.*SECOND*, 0);

Pada kode di atas, kita memecah data date dan time untuk mengambil nilai tahun, bulan, hari, jam dan menit.

* 1. calendar.set(Calendar.*MONTH*, Integer.*parseInt*(dateArray[1])-1);

Perlu menjadi perhatian mengapa kode di atas, data kita kurangi 1.

Misal tanggal yang kita masukkan adalah 2016-09-27. Jika kita pecah, maka kita akan memperoleh nilai 2016 (tahun), 9 (bulan), dan 27 (hari).

Masalahnya adalah, nilai bulan ke 9 pada kelas Calendar bukanlah bulan September. Ini karena indeksnya dimulai dari 0. Jadi, untuk memperoleh bulan September, maka nilai 9 tadi harus kita kurangi 1.

Intent yang dibuat akan dieksekusi ketika waktu alarm sama dengan waktu pada sistem Android. Di sini komponen PendingIntent  akan diberikan kepada BroadcastReceiver.

* 1. int requestCode = NOTIF\_ID\_ONETIME;
  2. PendingIntent pendingIntent =  PendingIntent.*getBroadcast*(context, requestCode, intent, 0);
  3. alarmManager.set(AlarmManager.*RTC\_WAKEUP*, calendar.getTimeInMillis(), pendingIntent);

Yang membedakan satu alarm dengan alarm lain adalah pada nilai requestCode.

* 1. int requestCode = NOTIF\_ID\_ONETIME;
  2. PendingIntent pendingIntent =  PendingIntent.*getBroadcast*(context, requestCode, intent, 0);

Jika kita merubah nilai waktu dan menjalankan ulang alarm dengan requestCode yang sama, maka akan merubah yang sudah diset sebelumnya.

Maksud dari baris ini alarmManager.set(AlarmManager.RTC\_WAKEUP, calendar.getTimeInMillis(), pendingIntent); adalah kita memasang alarm yang dibuat dengan tipe RTC\_WAKEUP. Tipe alarm ini dapat membangunkan peranti (jika dalam posisi sleep) untuk menjalankan obyek PendingIntent. Ketika kondisi sesuai, maka akan menjalankan BroadcastReceiver dengan semua proses yang terdapat didalam method onReceive().

* 1. String type = intent.getStringExtra(*EXTRA\_TYPE*);
  2. String message = intent.getStringExtra(*EXTRA\_MESSAGE*);
  3. String title = type.equalsIgnoreCase(*TYPE\_ONE\_TIME*) ? "One Time Alarm" : "Repeating Alarm";
  4. int notifId = type.equalsIgnoreCase(*TYPE\_ONE\_TIME*) ? NOTIF\_ID\_ONETIME : NOTIF\_ID\_REPEATING;
  5. showAlarmNotification(context, title, message, notifId);

Sebuah notifikasi sederhana akan tampil pada panel notifikasi pengguna dengan bunyi notifikasi umum dan getaran. Di sini metode showAlarmNotification(context, title, message, notifId); dijalankan.

* 1. private void showAlarmNotification(Context context, String title, String message, int notifId){
  2. NotificationManager notificationManagerCompat = (NotificationManager) context.getSystemService(Context.*NOTIFICATION\_SERVICE*);
  3. Uri alarmSound = RingtoneManager.*getDefaultUri*(RingtoneManager.*TYPE\_NOTIFICATION*);
  4. NotificationCompat.Builder builder = new NotificationCompat.Builder(context)
  5. .setSmallIcon(R.drawable.*ic\_access\_alarm\_black\_36dp*)
  6. .setContentTitle(title)
  7. .setContentText(message)
  8. .setColor(ContextCompat.*getColor*(context, android.R.color.*transparent*))
  9. .setVibrate(new long[]{1000, 1000, 1000, 1000, 1000})
  10. .setSound(alarmSound);
  11. notificationManagerCompat.notify(notifId, builder.build());
  12. }

Metode diatas merupakan sebuah method untuk membuat dan menampilkan notifikasi yang kompatibel dengan beragam API dari Android. Metode ini memanfaatkan fasilitas NotificationCompat. Anda akan lebih jauh memahami tentang Notifikasi ini pada modul berikutnya.

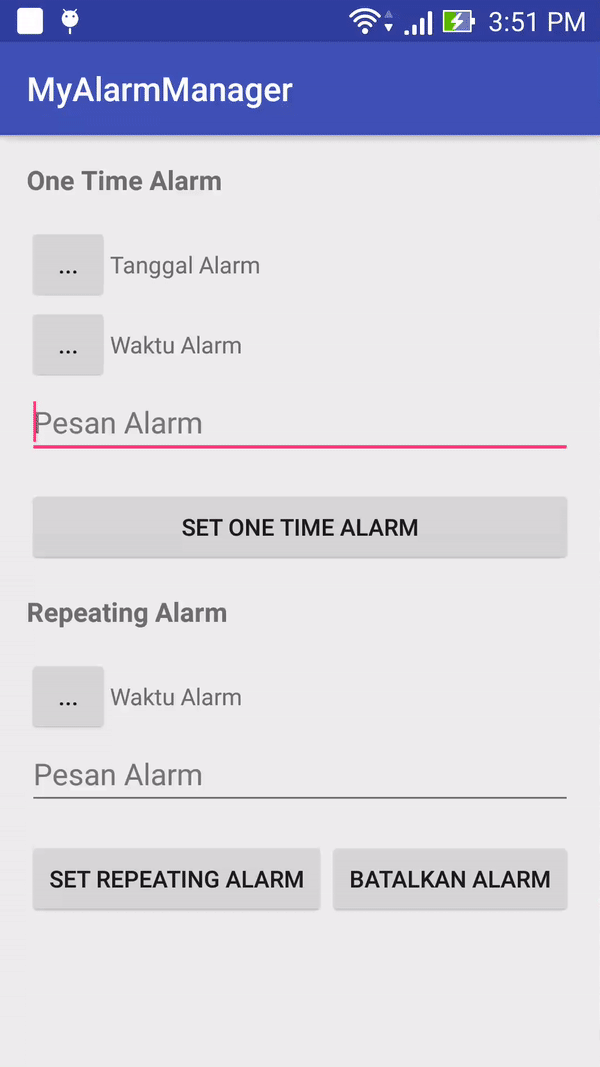
Baik, kami berharap Anda mendapatkan gambaran menyeluruh tentang ide, konsep dan skenario yang diimplementasikan pada bagian "menjalankan one-time-alarm manager".

Selanjutnya, kita akan membuat Repeating Alarm untuk melakukan tugas yang berulang. Jadi ayo kita ngoding lagi.

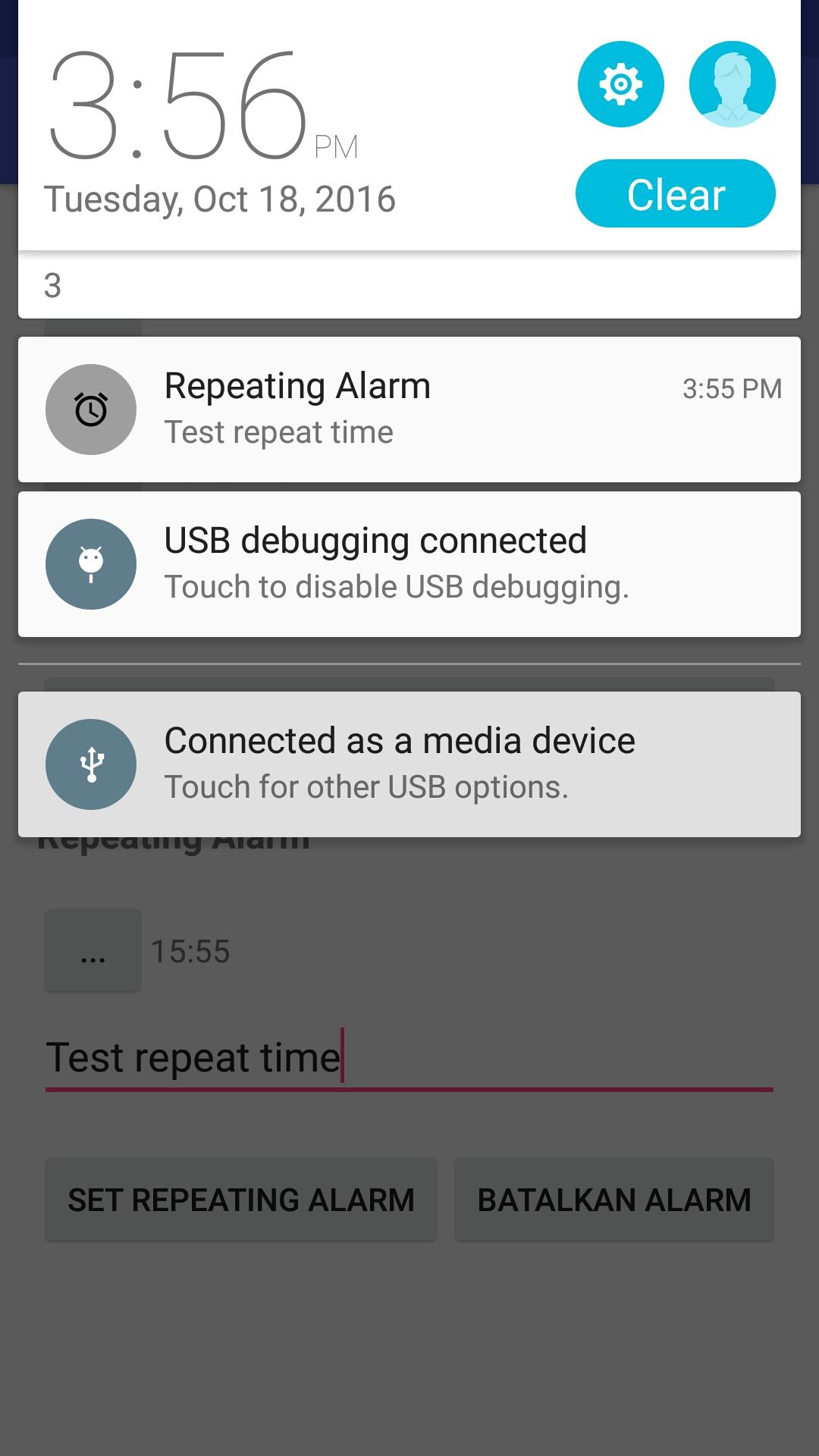
*From <*[*https://www.dicoding.com/academies/14/tutorials/189?from=187*](https://www.dicoding.com/academies/14/tutorials/189?from=187)*>*

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.  
   public void setRepeatingAlarm(Context context, String type, String time, String message){  
      AlarmManager alarmManager = (AlarmManager)context.getSystemService(Context.*ALARM\_SERVICE*);  
      Intent intent = new Intent(context, AlarmReceiver.class);  
      intent.putExtra(*EXTRA\_MESSAGE*, message);  
      intent.putExtra(*EXTRA\_TYPE*, type);  
      String timeArray[] = time.split(":");  
      Calendar calendar = Calendar.*getInstance*();  
      calendar.set(Calendar.*HOUR\_OF\_DAY*, Integer.*parseInt*(timeArray[0]));  
      calendar.set(Calendar.*MINUTE*, Integer.*parseInt*(timeArray[1]));  
      calendar.set(Calendar.*SECOND*, 0);  
      int requestCode = NOTIF\_ID\_ONETIME;  
      PendingIntent pendingIntent =  PendingIntent.*getBroadcast*(context, requestCode, intent, 0);  
      alarmManager.setInexactRepeating(AlarmManager.*RTC\_WAKEUP*, calendar.getTimeInMillis(), AlarmManager.*INTERVAL\_DAY*, pendingIntent);  
      Toast.*makeText*(context, "Repeating alarm set up", Toast.*LENGTH\_SHORT*).show();  
   }
2. Sehingga kode AlarmManager kita saat ini menjadi seperti ini.  
   public class AlarmReceiver extends BroadcastReceiver {  
       public static final String TYPE\_ONE\_TIME = "OneTimeAlarm";  
       public static final String TYPE\_REPEATING = "RepeatingAlarm";  
       public static final String EXTRA\_MESSAGE = "message";  
       public static final String EXTRA\_TYPE = "type";  
       private final int NOTIF\_ID\_ONETIME = 100;  
       private final int NOTIF\_ID\_REPEATING = 101;  
       public AlarmReceiver() {  
       }  
       @Override  
       public void onReceive(Context context, Intent intent) {  
           String type = intent.getStringExtra(EXTRA\_TYPE);  
           String message = intent.getStringExtra(EXTRA\_MESSAGE);  
           String title = type.equalsIgnoreCase(TYPE\_ONE\_TIME) ? "One Time Alarm" : "Repeating Alarm";  
           int notifId = type.equalsIgnoreCase(TYPE\_ONE\_TIME) ? NOTIF\_ID\_ONETIME : NOTIF\_ID\_REPEATING;  
           showAlarmNotification(context, title, message, notifId);  
       }  
       private void showAlarmNotification(Context context, String title, String message, int notifId){  
           NotificationManager notificationManagerCompat = (NotificationManager) context.getSystemService(Context.NOTIFICATION\_SERVICE);  
           Uri alarmSound = RingtoneManager.getDefaultUri(RingtoneManager.TYPE\_NOTIFICATION);  
           NotificationCompat.Builder builder = new NotificationCompat.Builder(context)  
                   .setSmallIcon(R.drawable.ic\_access\_alarm\_black\_36dp)  
                   .setContentTitle(title)  
                   .setContentText(message)  
                   .setColor(ContextCompat.getColor(context, android.R.color.transparent))  
                   .setVibrate(new long[]{1000, 1000, 1000, 1000, 1000})  
                   .setSound(alarmSound);  
           notificationManagerCompat.notify(notifId, builder.build());  
       }  
       public void setOneTimeAlarm(Context context, String type, String date, String time, String message){  
           AlarmManager alarmManager = (AlarmManager)context.getSystemService(Context.ALARM\_SERVICE);  
           Intent intent = new Intent(context, AlarmReceiver.class);  
           intent.putExtra(EXTRA\_MESSAGE, message);  
           intent.putExtra(EXTRA\_TYPE, type);  
           String dateArray[] = date.split("-");  
           String timeArray[] = time.split(":");  
           Calendar calendar = Calendar.getInstance();  
           calendar.set(Calendar.YEAR, Integer.parseInt(dateArray[0]));  
           calendar.set(Calendar.MONTH, Integer.parseInt(dateArray[1])-1);  
           calendar.set(Calendar.DAY\_OF\_MONTH, Integer.parseInt(dateArray[2]));  
           calendar.set(Calendar.HOUR\_OF\_DAY, Integer.parseInt(timeArray[0]));  
           calendar.set(Calendar.MINUTE, Integer.parseInt(timeArray[1]));  
           calendar.set(Calendar.SECOND, 0);  
           int requestCode = NOTIF\_ID\_ONETIME;  
           PendingIntent pendingIntent =  PendingIntent.getBroadcast(context, requestCode, intent, 0);  
           alarmManager.set(AlarmManager.RTC\_WAKEUP, calendar.getTimeInMillis(), pendingIntent);  
           Toast.makeText(context, "One time alarm set up", Toast.LENGTH\_SHORT).show();  
       }  
       **public void setRepeatingAlarm(Context context, String type, String time, String message){  
           AlarmManager alarmManager = (AlarmManager)context.getSystemService(Context.ALARM\_SERVICE);  
           Intent intent = new Intent(context, AlarmReceiver.class);  
           intent.putExtra(EXTRA\_MESSAGE, message);  
           intent.putExtra(EXTRA\_TYPE, type);  
           String timeArray[] = time.split(":");  
           Calendar calendar = Calendar.getInstance();  
           calendar.set(Calendar.HOUR\_OF\_DAY, Integer.parseInt(timeArray[0]));  
           calendar.set(Calendar.MINUTE, Integer.parseInt(timeArray[1]));  
           calendar.set(Calendar.SECOND, 0);  
           int requestCode = NOTIF\_ID\_REPEATING;  
           PendingIntent pendingIntent =  PendingIntent.getBroadcast(context, requestCode, intent, 0);  
           alarmManager.setInexactRepeating(AlarmManager.RTC\_WAKEUP, calendar.getTimeInMillis(), AlarmManager.INTERVAL\_DAY, pendingIntent);  
           Toast.makeText(context, "Repeating alarm set up", Toast.LENGTH\_SHORT).show();  
       }**  
       public void cancelAlarm(Context context, String type){  
           AlarmManager alarmManager = (AlarmManager)context.getSystemService(Context.ALARM\_SERVICE);  
           Intent intent = new Intent(context, AlarmReceiver.class);  
           int requestCode = type.equalsIgnoreCase(TYPE\_ONE\_TIME) ? NOTIF\_ID\_ONETIME : NOTIF\_ID\_REPEATING;  
           PendingIntent pendingIntent = PendingIntent.getBroadcast(context, requestCode, intent, 0);  
           alarmManager.cancel(pendingIntent);  
       }  
   }
3. Selanjutnya, pada method onClick() di MainActivity silakan tambahkan baris berikut :  
   else if (v.getId() == R.id.*btn\_repeating\_time\_alarm\_time*){  
       final Calendar currentDate = Calendar.*getInstance*();  
       new TimePickerDialog(this, new TimePickerDialog.OnTimeSetListener() {  
           @Override  
           public void onTimeSet(TimePicker view, int hourOfDay, int minute) {  
               calRepeatTimeTime.set(Calendar.*HOUR\_OF\_DAY*, hourOfDay);  
               calRepeatTimeTime.set(Calendar.*MINUTE*, minute);  
               SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm");  
               tvRepeatingTime.setText(dateFormat.format(calRepeatTimeTime.getTime()));}  
       }, currentDate.get(Calendar.*HOUR\_OF\_DAY*), currentDate.get(Calendar.*MINUTE*), true).show();  
   }  
   else if (v.getId() == R.id.*btn\_repeating\_time\_alarm*){  
       SimpleDateFormat timeFormat = new SimpleDateFormat("HH:mm");  
       String repeatTimeTime = timeFormat.format(calRepeatTimeTime.getTime());  
       String repeatTimeMessage = edtRepeatingMessage.getText().toString();  
     
       alarmPreference.setRepeatingTime(repeatingTimeTime);  
       alarmPreference.setRepeatingMessage(repeatingTimeMessage);  
     
       setRepeatingText();  
     
       alarmReceiver.setRepeatingAlarm(this, AlarmReceiver.*TYPE\_REPEATING*,  
               alarmPreference.getRepeatingTime(), alarmPreference.getRepeatingMessage());  
     
   }
4. Sehingga secara keseluruhan kode method onClick() kita menjadi seperti ini :  
   @Override  
       public void onClick(View v) {  
           if (v.getId() == R.id.*btn\_one\_time\_alarm\_date*){  
               final Calendar currentDate = Calendar.*getInstance*();  
               new DatePickerDialog(this, new DatePickerDialog.OnDateSetListener() {  
                   @Override  
                   public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {  
                       calOneTimeDate.set(year, monthOfYear, dayOfMonth);  
                       SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");  
                       tvOneTimeDate.setText(dateFormat.format(calOneTimeDate.getTime()));  
                   }  
               }, currentDate.get(Calendar.*YEAR*), currentDate.get(Calendar.*MONTH*), currentDate.get(Calendar.*DATE*)).show();  
           }  
           else if (v.getId() == R.id.*btn\_one\_time\_alarm\_time*){  
               final Calendar currentDate = Calendar.*getInstance*();  
               new TimePickerDialog(this, new TimePickerDialog.OnTimeSetListener() {  
                   @Override  
                   public void onTimeSet(TimePicker view, int hourOfDay, int minute) {  
                       calOneTimeTime.set(Calendar.*HOUR\_OF\_DAY*, hourOfDay);  
                       calOneTimeTime.set(Calendar.*MINUTE*, minute);  
                       SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm");  
                       tvOneTimeTime.setText(dateFormat.format(calOneTimeTime.getTime()));}  
               }, currentDate.get(Calendar.*HOUR\_OF\_DAY*), currentDate.get(Calendar.*MINUTE*), true).show();  
     
           }**else if (v.getId() == R.id.*btn\_repeating\_time\_alarm\_time*){  
               final Calendar currentDate = Calendar.*getInstance*();  
               new TimePickerDialog(this, new TimePickerDialog.OnTimeSetListener() {  
                   @Override  
                   public void onTimeSet(TimePicker view, int hourOfDay, int minute) {  
                       calRepeatTimeTime.set(Calendar.*HOUR\_OF\_DAY*, hourOfDay);  
                       calRepeatTimeTime.set(Calendar.*MINUTE*, minute);  
                       SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm");  
                       tvRepeatingTime.setText(dateFormat.format(calRepeatTimeTime.getTime()));  
                       *//Log.v(TAG, "The choosen one " + date.getTime());*}  
               }, currentDate.get(Calendar.*HOUR\_OF\_DAY*), currentDate.get(Calendar.*MINUTE*), true).show();  
           }**  
           else if (v.getId() == R.id.*btn\_set\_one\_time\_alarm*) {  
     
                   SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");  
                   String oneTimeDate = dateFormat.format(calOneTimeDate.getTime());  
                   SimpleDateFormat timeFormat = new SimpleDateFormat("HH:mm");  
                   String oneTimeTime = timeFormat.format(calOneTimeTime.getTime());  
                   String oneTimeMessage = edtOneTimeMessage.getText().toString();  
     
                   alarmPreference.setOneTimeDate(oneTimeDate);  
                   alarmPreference.setOneTimeMessage(oneTimeMessage);  
                   alarmPreference.setOneTimeTime(oneTimeTime);  
     
                   *//setOneTimeText();*alarmReceiver.setOneTimeAlarm(this, AlarmReceiver.*TYPE\_ONE\_TIME*,  
                           alarmPreference.getOneTimeDate(),  
                           alarmPreference.getOneTimeTime(),  
                           alarmPreference.getOneTimeMessage());  
     
           }  
           **else if (v.getId() == R.id.*btn\_repeating\_time\_alarm*){  
               SimpleDateFormat timeFormat = new SimpleDateFormat("HH:mm");  
     
               String repeatTimeTime = timeFormat.format(calRepeatTimeTime.getTime());  
               String repeatTimeMessage = edtRepeatingMessage.getText().toString();  
   alarmPreference.setRepeatingTime(repeatTimeTime);  
               alarmPreference.setRepeatingMessage(repeatTimeMessage);  
     
   alarmReceiver.setRepeatingAlarm(this, AlarmReceiver.*TYPE\_REPEATING*,  
                       alarmPreference.getRepeatingTime(), alarmPreference.getRepeatingMessage());  
     
           }**  
           else if (v.getId() == R.id.*btn\_cancel\_repeating\_alarm*){  
               alarmReceiver.cancelAlarm(this, AlarmReceiver.*TYPE\_REPEATING*);  
           }  
       }
5. Kemudian lengkapi dengan membuat method setRepeatingText() seperti berikut :  
   private void setRepeatingText() {  
       tvRepeatingTime.setText(alarmPreference.getRepeatingTime());  
       edtRepeatingMessage.setText(alarmPreference.getRepeatingMessage());  
   }
6. Silakan tambahkan baris berikut di method onCreate() yang dapat memberikan nilai default yang pernah kita gunakan ketika aplikasi kita buka  
   if (!TextUtils.*isEmpty*(alarmPreference.getRepeatingTime())){  
      setRepeatingText();  
   }
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.



*From <*[*https://www.dicoding.com/academies/14/tutorials/191*](https://www.dicoding.com/academies/14/tutorials/191)*>*

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!

Bedah Kode

Tidak ada perubahan yang signifikan ketika kita membuat metode setRepeatingAlarm() dengan method setOneTimeAlarm(). Yang membedakan adalah pada pemanggilan metode terkait. Pada setOneTimeAlarm() kita menggunakan metode set() sedangkan pada setRepeatingAlarm() kita menggunakan setInexactRepeating() .

public void setRepeatingAlarm(Context context, String type, String time, String message){  
   AlarmManager alarmManager = (AlarmManager)context.getSystemService(Context.*ALARM\_SERVICE*);  
   Intent intent = new Intent(context, AlarmReceiver.class);  
   intent.putExtra(*EXTRA\_MESSAGE*, message);  
   intent.putExtra(*EXTRA\_TYPE*, type);  
   String timeArray[] = time.split(":");  
   Calendar calendar = Calendar.*getInstance*();  
   calendar.set(Calendar.*HOUR\_OF\_DAY*, Integer.*parseInt*(timeArray[0]));  
   calendar.set(Calendar.*MINUTE*, Integer.*parseInt*(timeArray[1]));  
   calendar.set(Calendar.*SECOND*, 0);  
   int requestCode = NOTIF\_ID\_ONETIME;  
   PendingIntent pendingIntent =  PendingIntent.*getBroadcast*(context, requestCode, intent, 0);  
   alarmManager.setInexactRepeating(AlarmManager.*RTC\_WAKEUP*, calendar.getTimeInMillis(), AlarmManager.*INTERVAL\_DAY*, pendingIntent);  
   Toast.*makeText*(context, "Repeating alarm set up", Toast.*LENGTH\_SHORT*).show();  
}

Secara umum proses yang disusun pada metode diatas adalah sama. Yang membedakan hanya ide dan skenarionya. Disini kita akan mendemonstrasikan sebuah proses yang berulang dengan satuan waktu yang ditentukan dengan memanfaatkan fasilitas AlarmManager.

alarmManager.setInexactRepeating(AlarmManager.*RTC\_WAKEUP*, calendar.getTimeInMillis(), AlarmManager.*INTERVAL\_DAY*, pendingIntent);

Baris diatas akan menjalankan obyek PendingIntent di setiap waktu yang ditentukan dalam millissecond dengan interval per hari. Penggunaan metode setInexactRepeating() menjadi pilihan yang lebih tepat karena Android akan menjalankan alarm ini secara bersamaan dengan alarm lain. Meskipun waktu antara tiap alarm tersebut tidak sama persis.

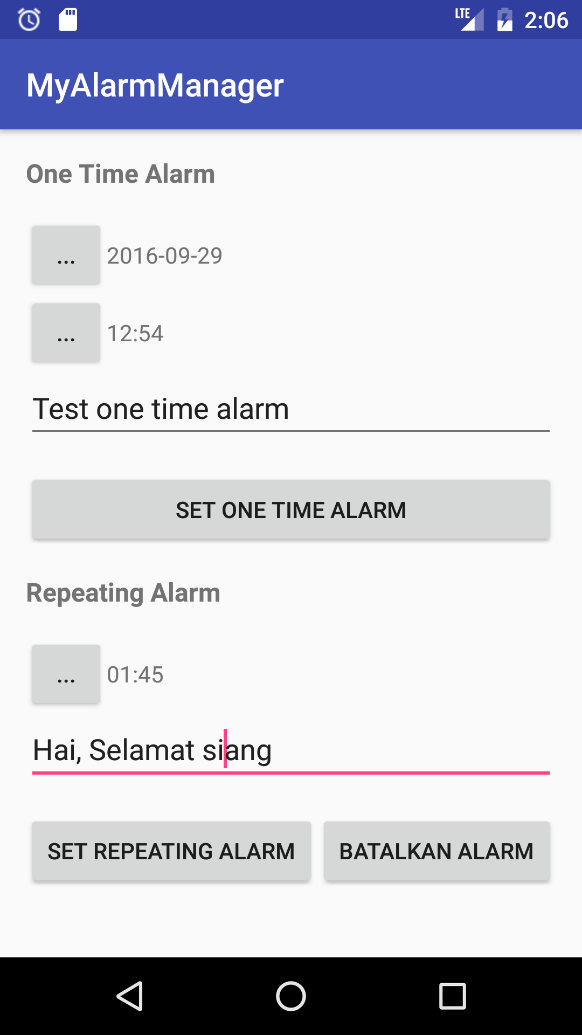
Ini penting dilakukan agar baterai device lebih hemat dan tidak cepat habis. Ingat! Penggunaan alarm manager yang tidak tepat akan memakan daya baterai yang cukup besar.

*From <*[*https://www.dicoding.com/academies/14/tutorials/191?from=189*](https://www.dicoding.com/academies/14/tutorials/191?from=189)*>*

Pada modul 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="Batalkan 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 ,btnRepeatingTime ,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\_repeating\_alarm*);
   3. btnCancelRepeatingAlarm.setOnClickListener(this);
3. Selanjutnya kita tambahkan kode seleksi baru pada method onClick() yang akan menjalankan proses pembatalan repeating alarm ketika tombol **Batalkan 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 ,btnRepeatingTime ,btnRepeating,  **btnCancelRepeatingAlarm**;
   10. private Calendar calOneTimeDate, calOneTimeTime,calRepeatTimeTime;
   12. private AlarmReceiver alarmReceiver;
   13. private AlarmPreference alarmPreference;
   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\_alarm\_date*);
   21. tvOneTimeTime = (TextView)findViewById(R.id.*tv\_one\_time\_alarm\_time*);
   22. edtOneTimeMessage = (EditText)findViewById(R.id.*edt\_one\_time\_alarm\_message*);
   23. btnOneTimeDate = (Button)findViewById(R.id.*btn\_one\_time\_alarm\_date*);
   24. btnOneTimeTime = (Button)findViewById(R.id.*btn\_one\_time\_alarm\_time*);
   25. btnOneTime = (Button)findViewById(R.id.*btn\_set\_one\_time\_alarm*);
   27. tvRepeatingTime = (TextView)findViewById(R.id.*tv\_repeating\_alarm\_time*);
   28. edtRepeatingMessage = (EditText)findViewById(R.id.*edt\_repeating\_alarm\_message*);
   29. btnRepeatingTime = (Button)findViewById(R.id.*btn\_repeating\_time\_alarm\_time*);
   30. btnRepeating = (Button)findViewById(R.id.*btn\_repeating\_time\_alarm*);
   32. **btnCancelRepeatingAlarm = (Button)findViewById(R.id.*btn\_cancel\_repeating\_alarm*);**
   34. btnOneTimeDate.setOnClickListener(this);
   35. btnOneTimeTime.setOnClickListener(this);
   36. btnOneTime.setOnClickListener(this);
   37. btnRepeatingTime.setOnClickListener(this);
   38. btnRepeating.setOnClickListener(this);
   39. **btnCancelRepeatingAlarm.setOnClickListener(this);**
   41. calOneTimeDate = Calendar.*getInstance*();
   42. calOneTimeTime = Calendar.*getInstance*();
   43. calRepeatTimeTime = Calendar.*getInstance*();
   45. alarmPreference = new AlarmPreference(this);
   46. alarmReceiver = new AlarmReceiver();
   48. if (!TextUtils.*isEmpty*(alarmPreference.getOneTimeDate())){
   49. setOneTimeText();
   50. }
   52. if (!TextUtils.*isEmpty*(alarmPreference.getRepeatingTime())){
   53. setRepeatingText();
   54. }
   56. }
   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.OnDateSetListener() {
   63. @Override
   64. public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {
   65. calOneTimeDate.set(year, monthOfYear, dayOfMonth);
   66. SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
   67. tvOneTimeDate.setText(dateFormat.format(calOneTimeDate.getTime()));
   68. }
   69. }, currentDate.get(Calendar.*YEAR*), currentDate.get(Calendar.*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.OnTimeSetListener() {
   74. @Override
   75. public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
   76. calOneTimeTime.set(Calendar.*HOUR\_OF\_DAY*, hourOfDay);
   77. calOneTimeTime.set(Calendar.*MINUTE*, minute);
   78. SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm");
   79. tvOneTimeTime.setText(dateFormat.format(calOneTimeTime.getTime()));
   80. }
   81. }, currentDate.get(Calendar.*HOUR\_OF\_DAY*), currentDate.get(Calendar.*MINUTE*), true).show();
   83. }else if (v.getId() == R.id.*btn\_repeating\_time\_alarm\_time*){
   84. final Calendar currentDate = Calendar.*getInstance*();
   85. new TimePickerDialog(this, new TimePickerDialog.OnTimeSetListener() {
   86. @Override
   87. public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
   88. calRepeatTimeTime.set(Calendar.*HOUR\_OF\_DAY*, hourOfDay);
   89. calRepeatTimeTime.set(Calendar.*MINUTE*, minute);
   90. SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm");
   91. tvRepeatingTime.setText(dateFormat.format(calRepeatTimeTime.getTime()));
   93. }
   94. }, currentDate.get(Calendar.*HOUR\_OF\_DAY*), currentDate.get(Calendar.*MINUTE*), true).show();
   95. }
   96. else if (v.getId() == R.id.*btn\_set\_one\_time\_alarm*) {
   98. SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
   100. String oneTimeDate = dateFormat.format(calOneTimeDate.getTime());
   102. SimpleDateFormat timeFormat = new SimpleDateFormat("HH:mm");
   104. String oneTimeTime = timeFormat.format(calOneTimeTime.getTime());
   105. String oneTimeMessage = edtOneTimeMessage.getText().toString();
   107. alarmPreference.setOneTimeDate(oneTimeDate);
   108. alarmPreference.setOneTimeMessage(oneTimeMessage);
   109. alarmPreference.setOneTimeTime(oneTimeTime);
   111. alarmReceiver.setOneTimeAlarm(this, AlarmReceiver.*TYPE\_ONE\_TIME*,
   112. alarmPreference.getOneTimeDate(),
   113. alarmPreference.getOneTimeTime(),
   114. alarmPreference.getOneTimeMessage());
   116. }
   117. else if (v.getId() == R.id.*btn\_repeating\_time\_alarm*){
   118. SimpleDateFormat timeFormat = new SimpleDateFormat("HH:mm");
   120. String repeatTimeTime = timeFormat.format(calRepeatTimeTime.getTime());
   121. String repeatTimeMessage = edtRepeatingMessage.getText().toString();
   122. alarmPreference.setRepeatingTime(repeatTimeTime);
   123. alarmPreference.setRepeatingMessage(repeatTimeMessage);

   126. alarmReceiver.setRepeatingAlarm(this, AlarmReceiver.*TYPE\_REPEATING*,
   127. alarmPreference.getRepeatingTime(), alarmPreference.getRepeatingMessage());
   129. }
   130. **else if (v.getId() == R.id.*btn\_cancel\_repeating\_alarm*){**
   131. **alarmReceiver.cancelAlarm(this, AlarmReceiver.*TYPE\_REPEATING*);**
   132. **}**
   134. }
   136. private void setOneTimeText() {
   137. tvOneTimeTime.setText(alarmPreference.getOneTimeTime());
   138. tvOneTimeDate.setText(alarmPreference.getOneTimeDate());
   139. edtOneTimeMessage.setText(alarmPreference.getOneTimeMessage());
   140. }
   142. private void setRepeatingText() {
   143. tvRepeatingTime.setText(alarmPreference.getRepeatingTime());
   144. edtRepeatingMessage.setText(alarmPreference.getRepeatingMessage());
   145. }
   146. }
4. Ketika aplikasi dijalankan, maka tampilannya kurang lebih menjadi seperti ini.



1. Terakhir, tambahkan sebuah Toast seperti berikut :
   1. Toast.*makeText*(context, "Repeating alarm dibatalkan", Toast.*LENGTH\_SHORT*).show();
2. 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.

Bedah Kode

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. }

Karena tipe alarm yang dikirimkan adalah TYPE\_REPEATING, maka nilai dari requestCode adalah NOTIF\_ID\_REPEATING yang bernilai 101. Setelah memperoleh requestCode tersebut, maka kita dapat mengetahui Intentnya. Dan kemudian kita dapat membatalkan alarmnya.

Dan itulah esensi membatalkan alarm yang telah dipasang.

Di materi selanjutnya kita akan berkenalan dengan yang lebih seru lagi, jobscheduler dan firebase dispatcher untuk melakukan proses *scheduling task* / proses berulang dengan cara yang lebih efisien dari sisi pemanfaatan sumber daya.

Untuk keseluruhan kode project ini bisa kamu unduh di

* [Source Code MyAlarmManager](https://github.com/dicodingacademy/a14-made-labs1/tree/master/07.Alarm/MyAlarmManager)

*From <*[*https://www.dicoding.com/academies/14/tutorials/193?from=141*](https://www.dicoding.com/academies/14/tutorials/193?from=141)*>*