Worksheet Praktikum 09: Notifikasi dan Alarm Manager

Mata Kuliah	Bahasa Pemrograman II (Praktikum)
Kode MK	S1085
Topik	Notifikasi dan Alarm
Pertemuan	8 (Delapan)
Dosen Pengampu	Afrig Aminuddin, S.Kom., M.Eng., Ph.D.
Alokasi Waktu	170 Menit (60 Menit Praktikum, 10 Menit Diskusi, 30 Menit Tugas, 60 Menit Mandiri)
Nama Mahasiswa	Revardi Maulaya Labib
NIM	24.12.3102

A. Tujuan Pembelajaran

Setelah menyelesaikan sesi praktikum ini, mahasiswa diharapkan:

- Mampu memahami konsep **Notification Channel** (wajib untuk Android Oreo ke atas).
- Mampu membuat dan menampilkan **Notifikasi** sederhana di status bar.
- Mampu memahami cara kerja **AlarmManager** untuk menjadwalkan tugas di masa depan.
- Mampu mengimplementasikan **BroadcastReceiver** untuk menangkap sinyal alarm dan memicu notifikasi.
- Mampu menangani izin (permission) `POST_NOTIFICATIONS` untuk Android 13 (Tiramisu).

B. Dasar Teori Singkat

- **Notification**: Pesan yang ditampilkan di luar UI aplikasi Anda (di bilah status) untuk memberikan pengingat, komunikasi dari orang lain, atau informasi tepat waktu lainnya. Sejak Android 8.0 (API 26), semua notifikasi harus dikelompokkan ke dalam **Notification Channel**.
- **AlarmManager**: Layanan sistem yang menyediakan akses ke layanan alarm sistem. Ini memungkinkan Anda menjadwalkan aplikasi Anda untuk dijalankan pada waktu tertentu di masa depan.
- **BroadcastReceiver**: Komponen Android yang memungkinkan aplikasi menerima pesan (broadcast) dari sistem Android atau aplikasi lain. Dalam kasus ini, AlarmManager akan mengirim sinyal broadcast yang akan ditangkap oleh Receiver untuk memunculkan notifikasi.

C. Alat dan Bahan

1. PC/Laptop dengan Android Studio.
2. Perangkat Android atau Emulator (Disarankan API 33/Android 13 untuk menguji permission).

D. Langkah-Langkah Praktikum (Guided Practice)

Tugas: Membuat Aplikasi "Simple Alarm" yang memunculkan Notifikasi pada jam yang ditentukan.

1. **Konfigurasi Izin (Manifest)**:
   - Buka AndroidManifest.xml.
   - Tambahkan izin berikut di atas tag <application>:

     XML

```xml
<uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
<uses-permission android:name="android.permission.SCHEDULE_EXACT_ALARM" />
```

2. **Membuat Layout (activity_main.xml)**:
   - Gunakan LinearLayout (vertikal).
   - Tambahkan komponen:
     - TimePicker (mode spinner atau clock) dengan id timePicker.
     - Button dengan id btn_set_alarm dan teks "Set Alarm".
     - TextView dengan id tv_status untuk menampilkan status alarm.
3. **Membuat BroadcastReceiver**:
   - Buat class Kotlin baru bernama AlarmReceiver.
   - Wariskan dari BroadcastReceiver.
   - Implementasikan metode onReceive. Di sinilah notifikasi akan dibuat.
   - **Penting:** Daftarkan receiver ini di AndroidManifest.xml di dalam tag <application>:

     XML

```xml
<receiver android:name=".AlarmReceiver" />
```

   - **Isi AlarmReceiver.kt**:

     Kotlin

```kotlin
class AlarmReceiver : BroadcastReceiver() {
    override fun onReceive(context: Context, intent: Intent) {
        // Logika memunculkan notifikasi
        val channelId = "alarm_channel"
        val channelName = "Alarm Channel"

        val notificationManager =
context.getSystemService(Context.NOTIFICATION_SERVICE) as
NotificationManager

        // Buat Channel (Wajib untuk Android O+)
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            val channel = NotificationChannel(channelId, channelName,
NotificationManager.IMPORTANCE_HIGH)
            notificationManager.createNotificationChannel(channel)
        }

        // Builder Notifikasi
        val builder = NotificationCompat.Builder(context, channelId)
```

```
            .setSmallIcon(R.drawable.ic_launcher_foreground) // Ganti dengan ikon
anda
            .setContentTitle("Alarm Berbunyi!")
            .setContentText("Waktunya untuk melakukan aktivitas Anda.")
            .setPriority(NotificationCompat.PRIORITY_HIGH)
            .setAutoCancel(true)

        // Tampilkan
        notificationManager.notify(123, builder.build())
    }
}
```

4. **Logika Utama (MainActivity.kt)**:
    - Buka MainActivity.kt.
    - Inisialisasi komponen UI.
    - Di dalam onCreate, buat logika tombol "Set Alarm".
    - Gunakan Calendar untuk mengambil waktu dari TimePicker.
    - Gunakan AlarmManager untuk menjadwalkan.

Kotlin

```
// Potongan kode di dalam setOnClickListener tombol
val calendar = Calendar.getInstance()
calendar.set(Calendar.HOUR_OF_DAY, timePicker.hour)
calendar.set(Calendar.MINUTE, timePicker.minute)
calendar.set(Calendar.SECOND, 0)

val alarmManager = getSystemService(Context.ALARM_SERVICE) as AlarmManager
val intent = Intent(this, AlarmReceiver::class.java)

// PendingIntent FLAG_IMMUTABLE wajib untuk Android 12+
val pendingIntent = PendingIntent.getBroadcast(this, 0, intent,
PendingIntent.FLAG_IMMUTABLE)

// Set Alarm
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
    alarmManager.setExactAndAllowWhileIdle(AlarmManager.RTC_WAKEUP,
calendar.timeInMillis, pendingIntent)
} else {
    alarmManager.setExact(AlarmManager.RTC_WAKEUP, calendar.timeInMillis,
pendingIntent)
}

Toast.makeText(this, "Alarm diatur!", Toast.LENGTH_SHORT).show()
```

5. **Handling Runtime Permission (Android 13+)**:
    - Di MainActivity, tambahkan pengecekan izin notifikasi saat aplikasi dibuka. Jika belum diizinkan, minta izin (requestPermissions).

E. Latihan Mandiri (Tugas)

Kerjakan tugas berikut untuk menguji pemahaman Anda:

1. **Kirim Pesan Kustom**:
    - Tambahkan sebuah EditText di activity_main.xml untuk menulis pesan pengingat (Misal: "Minum Obat").
    - Kirim teks tersebut menggunakan intent.putExtra("PESAN", ...) saat mengatur alarm.

- Di AlarmReceiver, tangkap pesan tersebut dengan intent.getStringExtra("PESAN") dan tampilkan sebagai ContentText pada notifikasi.

2. **Batalkan Alarm**:
   - Tambahkan tombol "Cancel Alarm".
   - Implementasikan logika alarmManager.cancel(pendingIntent) untuk membatalkan alarm yang sudah dijadwalkan.

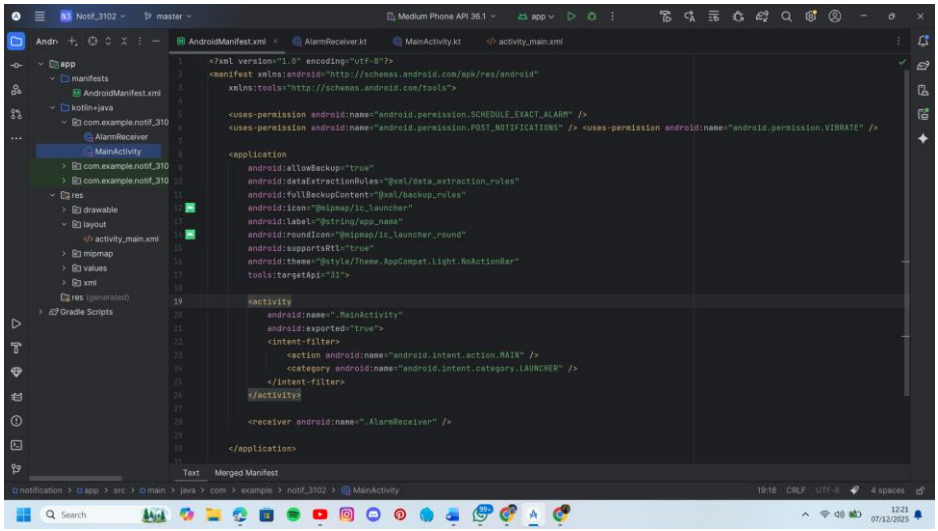
F. Kriteria Penilaian

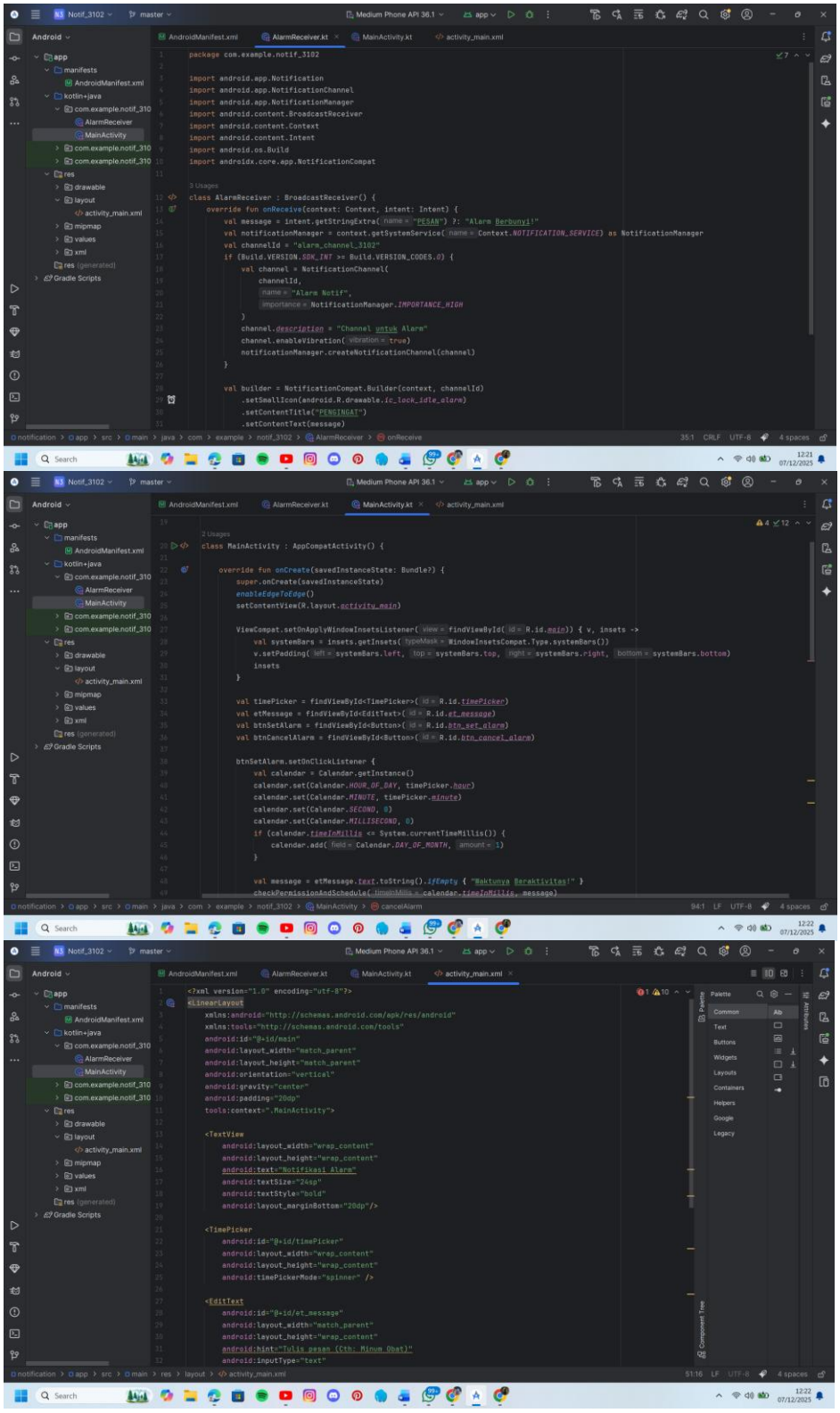
- **Izin & Manifest**: POST_NOTIFICATIONS dan receiver terdaftar dengan benar di Manifest.
- **Notification Channel**: Berhasil membuat channel notifikasi (aplikasi tidak crash di Android O+).
- **Fungsi Alarm**: Alarm berhasil memicu notifikasi muncul tepat waktu sesuai input TimePicker.
- **Latihan Mandiri**: Pesan kustom berhasil muncul di notifikasi dan fitur Cancel berfungsi.

G. Instruksi Pengumpulan

- *Commit* dan *Push* hasil pekerjaan Anda ke repository Git yang telah ditentukan.
- Beri nama *commit*: feat: implement notification and alarm manager.
- Batas waktu pengumpulan adalah sebelum pertemuan praktikum berikutnya.
- Lengkapi identitas dalam worksheet ini, kemudian kumpulkan ke Google Form yang disediakan (https://forms.gle/YgS6KKMfRNGdC1FKA).

Lembar Jawab:
Screenshot:





Link GitHub: https://github.com/myNameRev/Notif_3102.git