

Nathasya Eliora

1706979404

TKTPL B

Tugas UTS

Today

1. Menerapkan seluruh stack Android Framework standard

a. Menerapkan activity

Disini activitynya adalah main activity. Di main activity ini ada timer dan task.

Berisi:

- Inisialisasi variable
- Finder komponen ui
- Setting listener komponen

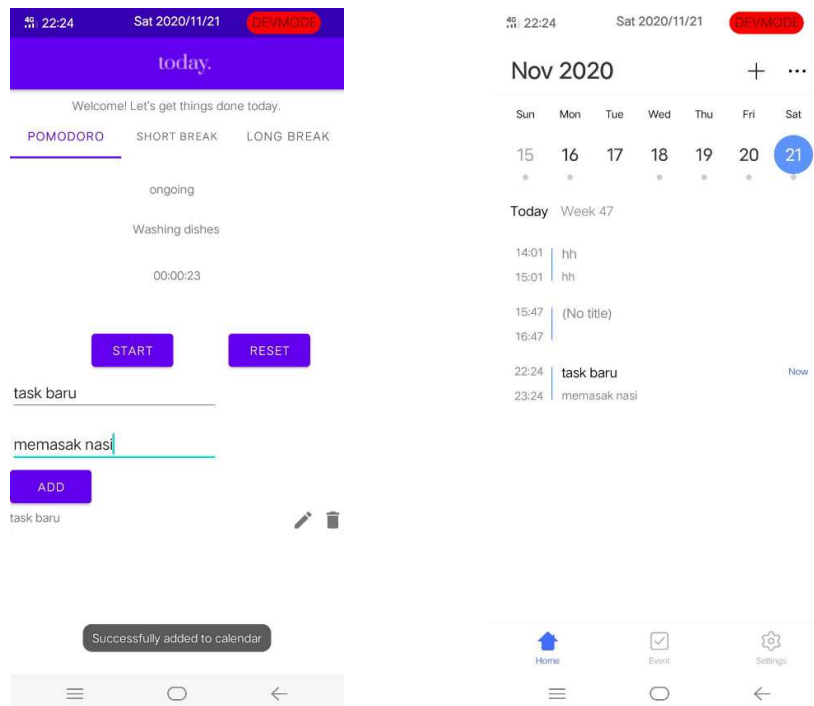
b. Menggunakan Service dan pemanggilan Remote Method (di luar aplikasi)

Service yang digunakan adalah penghitungan untuk alarm. Jadi ketika dia memilih sesi pomodoro (misal), akan di set alarm yang akan berbunyi +25menit setelahnya.jangan lupa mendaftarkan dulu di manifest.

```
@RequiresApi(Build.VERSION_CODES.KITKAT)
fun startAlarm(duration: Long) {
    val millis :Long = SystemClock.uptimeMillis() + duration
    val alarmManager :AlarmManager = getSystemService(Context.ALARM_SERVICE) as AlarmManager
    val intent = Intent( packageContext: this, AlarmReceiver::class.java)
    val pendingIntent :PendingIntent! = PendingIntent.getBroadcast( context: this, requestCode: 0, intent, flags: 0)
    alarmManager.setExact(AlarmManager.RTC_WAKEUP, millis, pendingIntent)
}
```

c. Memanfaatkan Content Provider

Menggunakan Calendar Provider ketika ada task baru akan langsung masuk ke calendar hp. Caranya adalah dengan mendaftarkan permission, lalu membuat fungsi di activity setiap tombol add button ditekan.



```

fun addToCalendar(tasktitle: String, taskdetail: String){
    var cr: ContentResolver? = this.getContentResolver()
    var cv: ContentValues = ContentValues()
    cv.put(CalendarContract.Events.TITLE, tasktitle)
    cv.put(CalendarContract.Events.DESRIPTION, taskdetail)
    cv.put(CalendarContract.Events.DTSTART, Calendar.getInstance().getTimeInMillis());
    cv.put(
        CalendarContract.Events.DTEND,
        Calendar.getInstance().getTimeInMillis() + 60 * 60 * 1000
    );
    cv.put(CalendarContract.Events.CALENDAR_ID, 1)
    cv.put(CalendarContract.Events.EVENT_TIMEZONE, Calendar.getInstance().getTimeZone().getID())
    var uri: Uri = cr?.insert(CalendarContract.Events.CONTENT_URI, cv)!!

    Toast.makeText(context: this, text: "Successfully added to calendar", Toast.LENGTH_SHORT).show()
}

```

d. Menerapkan BroadcastReceiver

Untuk menampilkan notifikasi ketika mencapai waktu yang telah ditentukan.

Caranya adalah membuat dahulu kelas broadcastreceivernya (untuk mengatur onreceive), membuat notificationutil (untuk bicara pada notification managernya), dan akhirnya dipanggil dengan method di activity.

```

class AlarmReceiver : BroadcastReceiver() {

    override fun onReceive(context: Context, intent: Intent) {
        // This method is called when the BroadcastReceiver is receiving an Intent broadcast.
        val notificationUtils = NotificationUtils(context)
        val notification : Notification! = notificationUtils.getNotificationBuilder().build()
        notificationUtils.getManager().notify( id: 150, notification)
    }
}

```



- e. Menerapkan Background Process (cth: AsyncTask) yang tidak “mati” ketika activity tidak aktif
Background process dilakukan dengan menggunakan runnable. Stopwatch tetap berjalan walau sedang tidak dibuka appnya.

```

val runnable = object : Runnable {

    override fun run() {
        if (boolstart == false) {
            boolstart = true
            display.text = "Pause"
        } else {
            boolstart = false
            display.text = "Resume"
        }

        seconds = SystemClock.uptimeMillis() / 1000 + pauseTime - startTime
        pause = seconds
        minutes = seconds / 60
        seconds = seconds % 60
        hours = hours / 60
        minutes = minutes % 60
    }
}

```

2. Menerapkan multi environment

a. Multi Layout

Ada 4, yaitu handphone screen portrait dan landscape, lalu ada tablet portrait dan landscape. Portrait di hp dan tablet sama, yaitu vertical linear. Sedangkan yang landscape dibagi dua kanan kiri.



today.

Welcome! Let's get things done today.

☐ Pomodoro ☐ Short Break ☐ Long Break

ongoing

00:00:00

START

RESET

Task Title

Task Detail

ADD

Item 0

Item 1

Item 2

Item 3

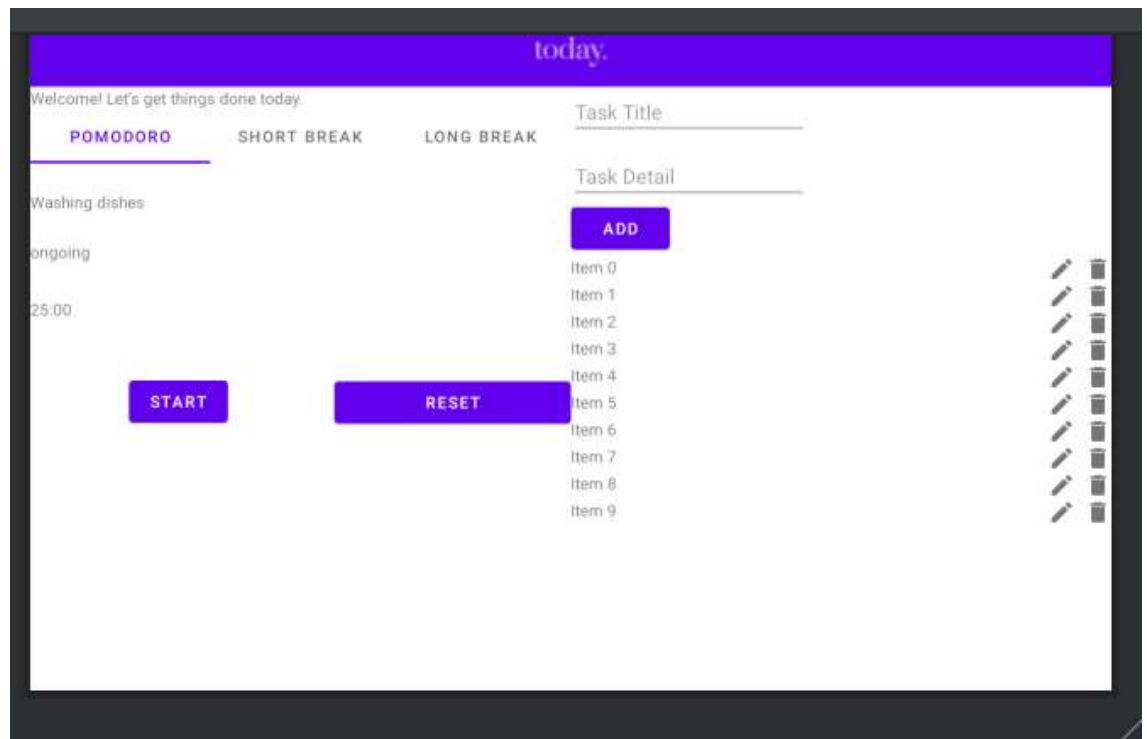
Item 4

Item 5

Item 6

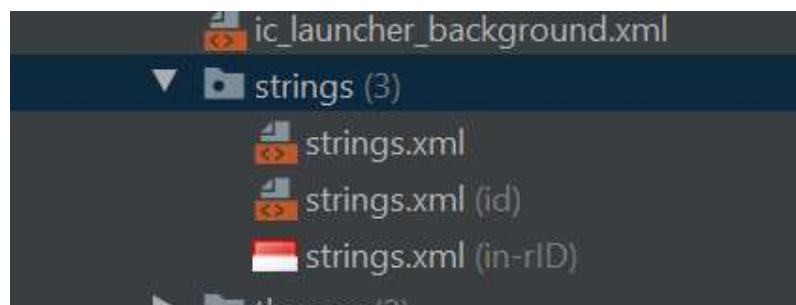
Item 7





b. Multi Language (i18n)

Dibuat string English dan indo. Saya melakukannya dengan memencet Bahasa di section kanan desain xml lalu create new language. Lalu dari situ diisi terjemahannya satu-satu. Sebelumnya semua string telah diextract dan diberi id supaya mudah untuk direplace.



3. Menerapkan Design Pattern MVVM & Background Task

Saya membuat file view model ketika dibutuhkan. Contohnya saat mengakses task dan log.

```

class MainViewModel(application: Application) : AndroidViewModel(application) {

    private var taskRepository = TaskRepository(application)
    private var tasks: LiveData<List<Task>>? = taskRepository.getTasks()

    fun insertTask(task: Task) {
        taskRepository.insert(task)
    }

    fun getTasks(): LiveData<List<Task>>? {
        return tasks
    }
}

```

Binding dilakukan dengan membuat adapter, lalu ditampilkan di recyclerview. Diobserve juga di viewmodel.

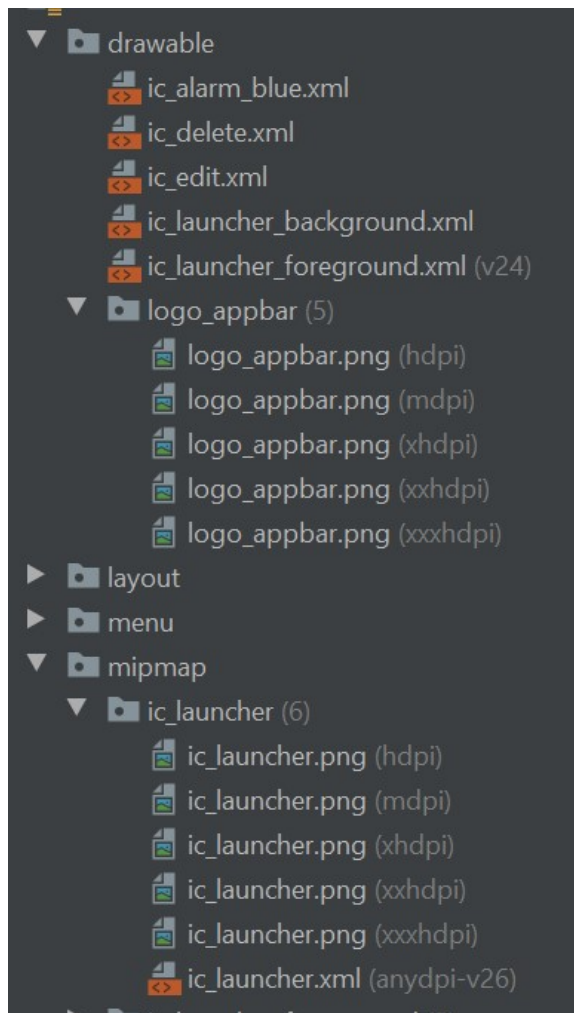
4. Menerapkan Assets dengan benar

a. String resource

Seperti di poin 2b

b. Mempersiapkan drawable resource untuk 2 (dua) screen size (& beda resolusi).

Saya mempersiapkan foto yang akan digunakan seperti logo dan clipart. Lalu file > new > asset. Lalu isi settingsnya dan akan otomatis dibuatkan untuk berbagai density.

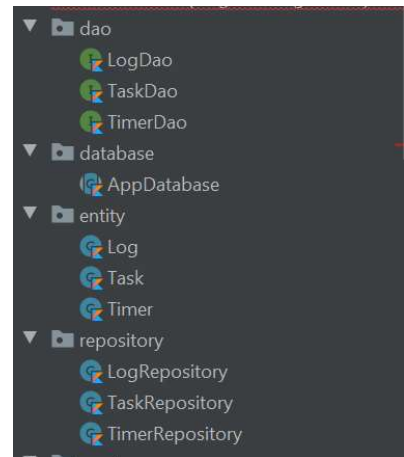


c. Memanfaatkan ContentProvider

Sudah di poin 1

5. Menerapkan Data Persistence

Data persistence diterapkan dengan menggunakan room. Pertama semua kebutuhan dimasukan ke gradle. Lalu mulai membuat entity, ada log, task, dan timer. Log menyimpan session yang dilakukan, jadi tiap pencet start, akan menyimpan log. Selanjutnya task untuk menyimpan task hari itu dan timer untuk menyimpan jenis2 sesi.



6. Inovasi

Saya menambahkan fitur split screen. Caranya adalah dengan memasukkan beberapa settings ke manifest.

```
<activity
    android:name=".MainActivity"
    android:resizeableActivity="true"
    android:label="Today"
    android:theme="@style/Theme.Today.NoActionBar">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>

    <layout android:defaultHeight="500dp"
        android:defaultWidth="600dp"
        android:gravity="top|end"
        android:minHeight="450dp"
        android:minWidth="300dp" />
</activity>
```