

Nama : Aries Wahyu Syaputra

NIM : D121171528

## Tugas Final Pemrograman Aplikasi Mobile

### Dokumentasi ToDo List App

Aplikasi ToDo List adalah aplikasi yang dapat menyimpan kegiatan/aktivitas yang akan dilakukan. Berikut dokumentasi alur aplikasi dan penggunaannya.

1. Menambah library pada **build.gradle** dan menambah apply plugin seperti dibawah ini:

```
1  apply plugin: 'kotlin-kapt'
2  apply plugin: 'kotlin-android-extensions'
3  apply plugin: 'com.android.application'
4  apply plugin: 'kotlin-android'
```

```
30 dependencies {
31     implementation fileTree(dir: 'libs', include: ['*.jar'])
32     implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
33     implementation 'androidx.appcompat:appcompat:1.1.0'
34     implementation 'androidx.core:core-ktx:1.2.0'
35     implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
36     implementation 'com.google.android.material:material:1.1.0'
37     testImplementation 'junit:junit:4.12'
38     androidTestImplementation 'androidx.test.ext:junit:1.1.1'
39     androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
40     // RecyclerView
41     implementation "androidx.recyclerview:recyclerview:1.1.0"
42     // coroutines
43     implementation "org.jetbrains.kotlinx:kotlinx-coroutines-android:1.3.2"
44     // Room
45     implementation "androidx.room:room-runtime:2.2.5"
46     implementation "androidx.room:room-ktx:2.2.5"
47     kapt "androidx.room:room-compiler:2.2.5"
48     // ViewModel
49     implementation "androidx.lifecycle:lifecycle-extensions:2.2.0"
50     kapt "androidx.lifecycle:lifecycle-compiler:2.2.0"
51 }
```

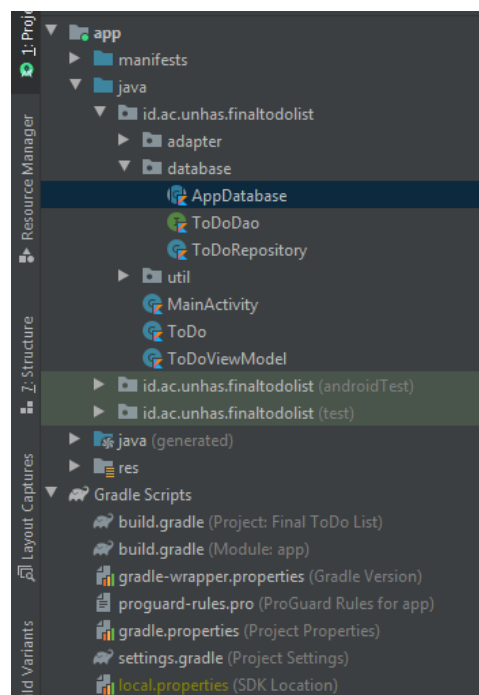
2. Men-sync **build.gradle** agar dapat mengambil dan menambahkan library dari online.

```
build.gradle (app) x AppDatabase.kt x ToDo.kt x activity_main.xml x ic_search.xml x menu_main.xml x ic_sort.xml x styles.xml
Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly. Sync Now
1  apply plugin: 'kotlin-kapt'
2  apply plugin: 'kotlin-android-extensions'
3  apply plugin: 'com.android.application'
4  apply plugin: 'kotlin-android'
5
6  android {
7      compileSdkVersion 29
8      buildToolsVersion "29.0.3"
9
10     defaultConfig {
11         applicationId "id.ac.unhas.finaltodolist"
12         minSdkVersion 21
13         targetSdkVersion 29
14         versionCode 1
15         versionName "1.0"
16
17         testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
18     }
19 }
```

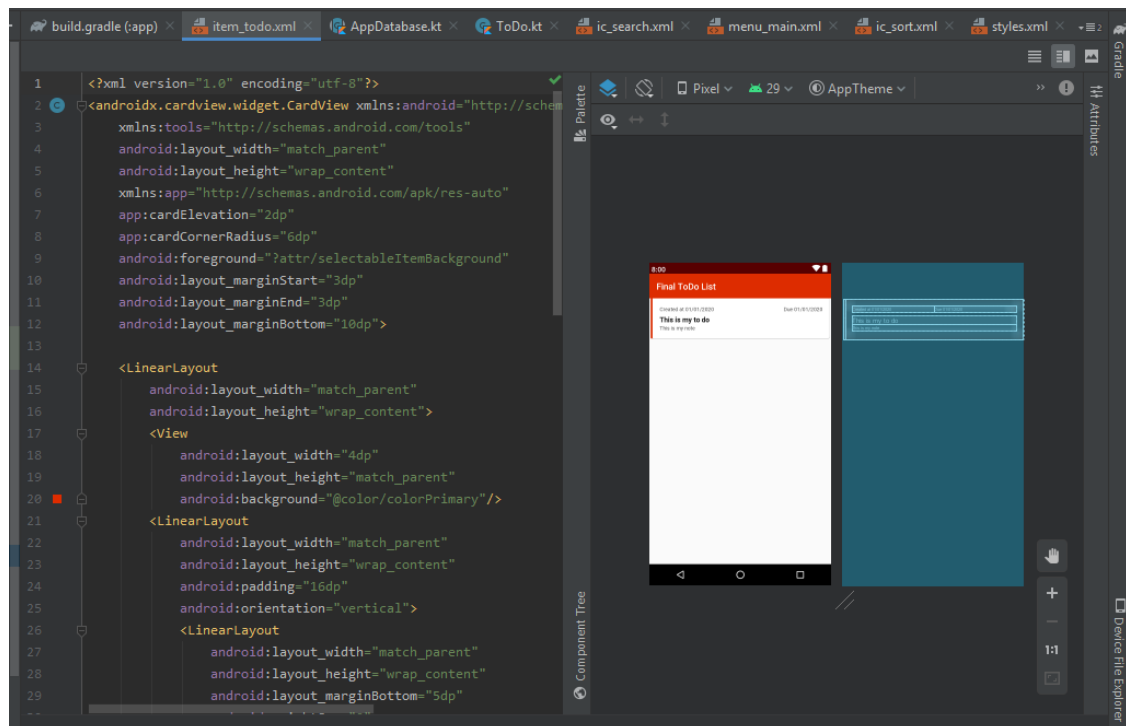
3. Membuat entity yang dapat merepresentasikan tabel dalam database. Entity ini dibuat dalam file **ToDo.kt** yang akan menjadi model data.

```
build.gradle (app) x AppDatabase.kt x ToDo.kt x activity_main.xml x ic_search.xml x menu_main.xml x
1 package id.ac.unhas.finaltodolist
2
3 import ...
4
5
6
7 //Model Data
8 @Entity(tableName = "todo_table")
9 data class ToDo(
10     @PrimaryKey(autoGenerate = true)
11     @ColumnInfo(name = "id")
12     val id: Int? = null,
13
14     @ColumnInfo(name = "title")
15     var title: String,
16
17     @ColumnInfo(name = "note")
18     var note: String? = null,
19
20     @ColumnInfo(name = "date_created")
21     var dateCreated: String,
22
23     @ColumnInfo(name = "date_updated")
24     var dateUpdated: String,
25
26     @ColumnInfo(name = "due_date")
27     var dueDate: String,
28
29     @ColumnInfo(name = "due_time")
30     var dueTime: String,
31
32     @ColumnInfo(name = "remind_me")
33     var remindMe: Boolean = true
34 )
```

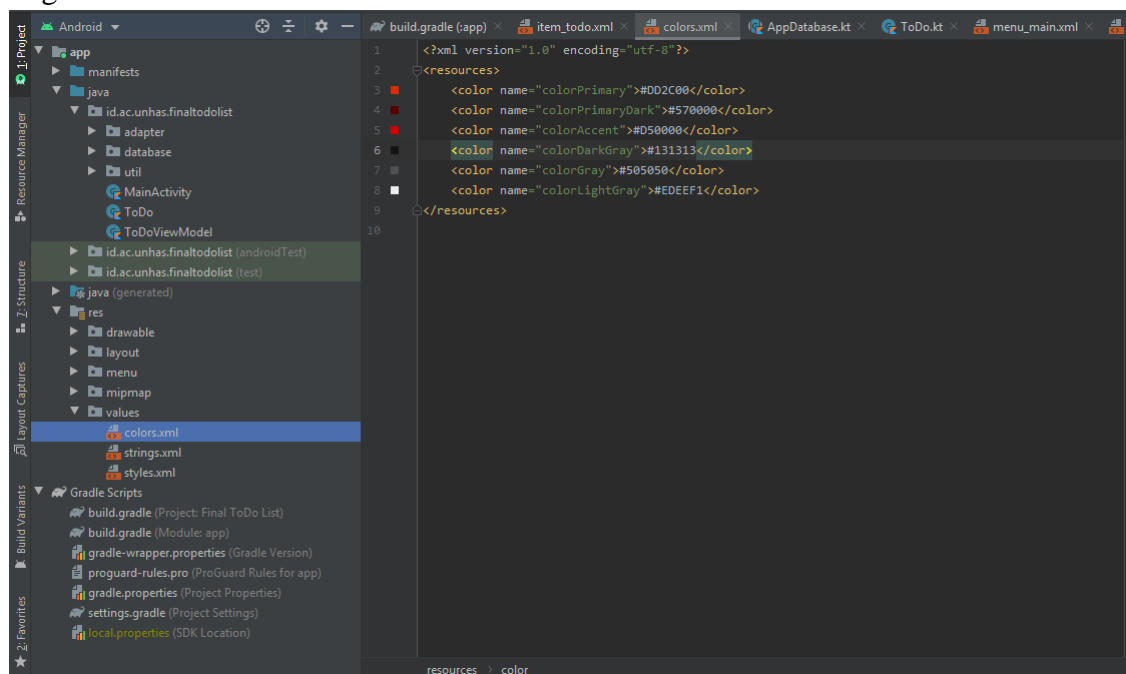
4. Membuat database dengan 3 file yaitu: **AppDatabase** (tempat pendefinisian dan titik akses untuk terhubung dengan database yang digunakan), **ToDoDao** (Database Access Object), **ToDoRepository** (penyimpanan) dan menyesuaikan parameter class dari Model Data.



5. Membuat layout model datanya dengan file **item\_todo.xml** yang akan mengatur tampilan dasar dari aplikasinya



6. Memodifikasi warna dari layout yang ada pada folder res/values/**color.xml** sesuai dengan keinginan.



7. Melakukan git push pertama sekaligus membuat repo pada github untuk fitur ToDo itemnya.

8. Membuat fitur CRUD, pada fungsi insert, delete, dan update merupakan proses I/O yang dapat memakan resource device pengguna. Fitur CRUD ini terletak pada file **ToDoViewModel** yang akan mengatur alur data yang tampil pada UI atau aplikasi yang dibuat.

```
ToDoViewModel.kt
build.gradle (app)
item_todo.xml
colors.xml
AppDatabase.kt
ToDo.kt

3  import android.app.Application
4  import androidx.lifecycle.AndroidViewModel
5  import androidx.lifecycle.LiveData
6  import id.ac.unhas.finaltodolist.database.ToDoRepository
7
8  class ToDoViewModel(application: Application): AndroidViewModel(application) {
9      private var todoRepository =
10         ToDoRepository(application)
11      private var todos: LiveData<List<ToDo>>? = todoRepository.todos()
12
13
14      fun getTodos() : LiveData<List<ToDo>>? {
15          return todos
16      }
17
18      fun insertToDo(todo: ToDo) {
19          todoRepository.insert(todo)
20      }
21
22      fun deleteToDo(todo: ToDo) {
23          todoRepository.delete(todo)
24      }
25
26      fun updateToDo(todo: ToDo) {
27          todoRepository.update(todo)
28      }
29  }
```

9. Membuat ToDoAdapter yang dapat mengatur tampilan pada list ToDo

```
app
item_todo.xml
colors.xml
AppDatabase.kt
ToDo.kt
styles.xml
FormDialog.kt
ToDoAdapter.kt

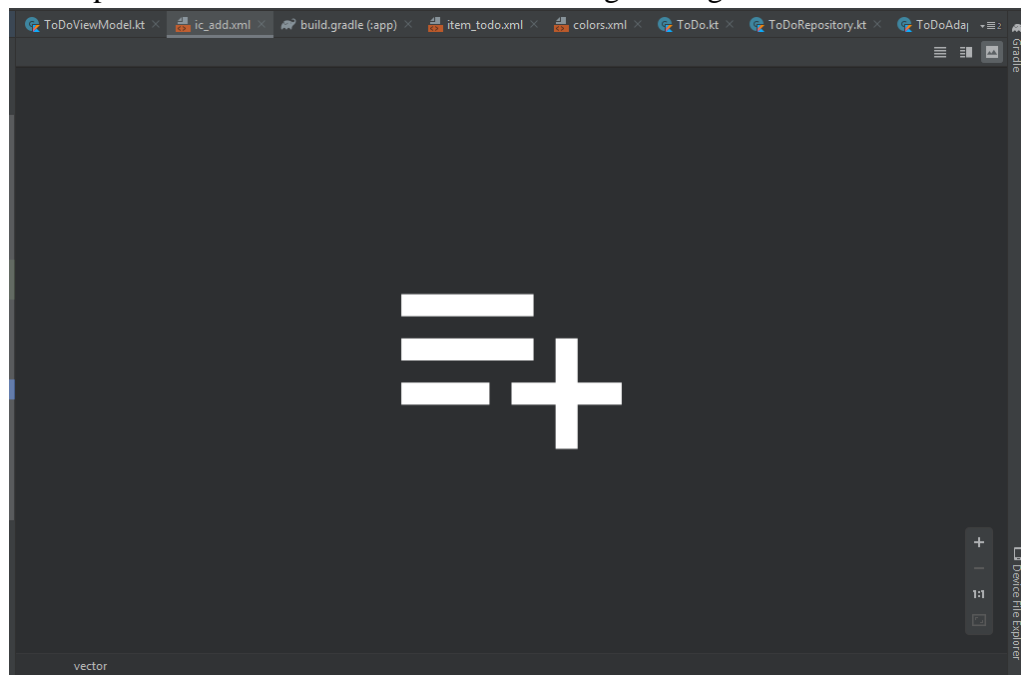
1  package id.ac.unhas.finaltodolist.adapter
2
3  import androidx.recyclerview.widget.RecyclerView
4  import androidx.recyclerview.widget.RecyclerView.ViewHolder
5
6  //mengatur tampilan list to do
7
8  class ToDoAdapter(private val listener: (ToDo, Int) -> Unit): RecyclerView.Adapter<RecyclerView.ViewHolder>() {
9      private val VIEW_TYPE_EMPTY = 0
10     private val VIEW_TYPE_TODO = 1
11
12     private var todoList = listOf<ToDo>()
13     private var todoFilteredList = listOf<ToDo>()
14
15     fun setToDoList(todoList: List<ToDo>) {
16         this.todoList = todoList
17         todoFilteredList = todoList
18         notifyDataSetChanged()
19     }
20 }
```

10. Membuat file Commons yang dapat memanggil fungsi-fungsi dari dialog

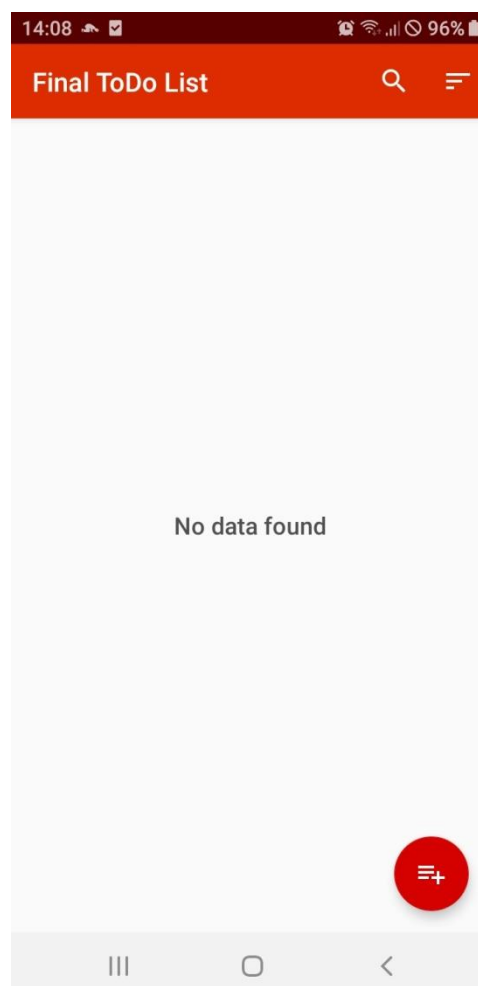
```
item_todo.xml
colors.xml
AppDatabase.kt
ToDo.kt
styles.xml
FormDialog.kt
ToDo.kt

8  import android.widget.EditText
9  import java.text.SimpleDateFormat
10 import java.util.*
11
12 //fungsi-fungsi yang dipanggil
13 object Commons {
14
15     fun showDatePickerDialog(context: Context, view: EditText) {
16         val calendar: Calendar = Calendar.getInstance()
17         DatePickerDialog(
18             context,
19             DatePickerDialog.OnDateSetListener { _, year, month, day ->
20                 calendar.set(Calendar.YEAR, year)
21                 calendar.set(Calendar.MONTH, month)
22                 calendar.set(Calendar.DAY_OF_MONTH, day)
23                 val myFormat = "dd/MM/yyyy"
24                 val sdf = SimpleDateFormat(myFormat, Locale.US)
25                 view.setText(sdf.format(calendar.time))
26             },
27             calendar.get(Calendar.YEAR),
28             calendar.get(Calendar.MONTH),
29             calendar.get(Calendar.DAY_OF_MONTH)
30         )
31         .show()
32     }
```

11. Membuat icon add pada folder `res/drawable/ic_add.xml` yang berfungsi menambahkan list. Kita dapat memilih icon sesuai kemauan masing-masing.



12. Membuat layout `item_empty.xml` untuk memastikan data list kosong atau tidak tersedia.



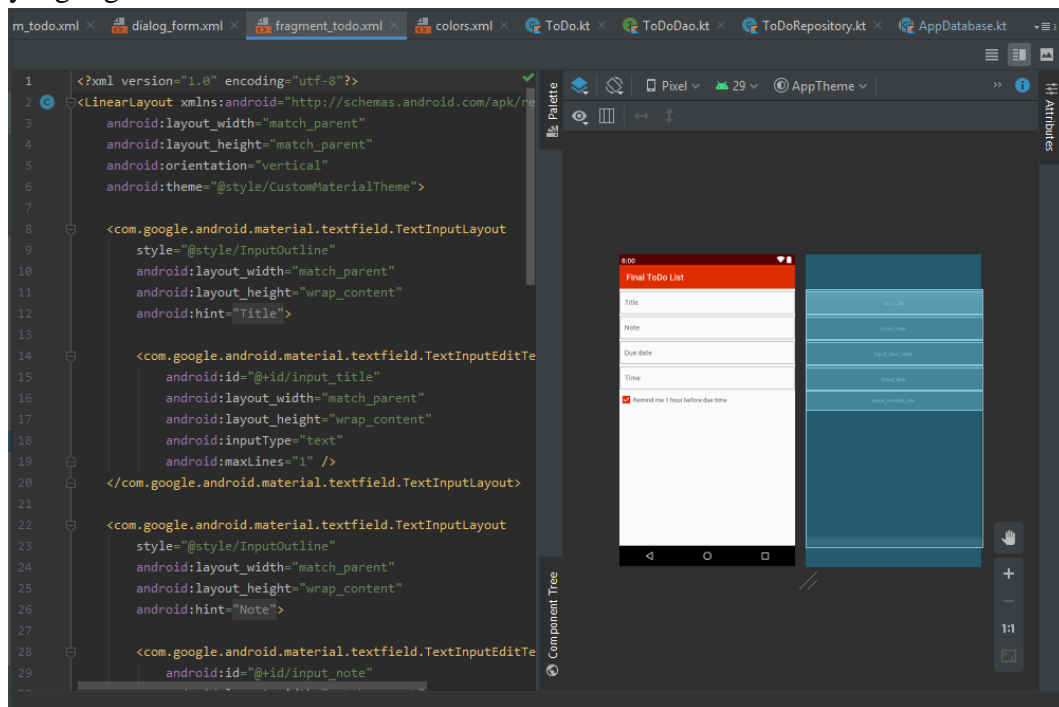
13. Menyesuaikan file string dengan data yang telah ditambahkan.
14. Menyesuaikan file yang telah ditambahkan pada **MainActivity** agar dapat menampilkan dan mengatur halaman aplikasi yang telah ditambahkan.
15. Membuat file **FormDialog** yang menjadi bentuk dialog interface dengan user.

```
7 import android.view.ViewGroup
8 import android.view.Window
9 import id.ac.unhas.finahtodolist.R
10 import kotlinx.android.synthetic.main.dialog_form.*
11
12
13 class FormDialog(context: Context, private val title: String, private val formLayout: View, private val saveAction: () -> Unit): Dialog(context, R.style.Theme_Dialog) {
14
15
16 override fun onCreate(savedInstanceState: Bundle?) {
17     super.onCreate(savedInstanceState)
18     requestWindowFeature(Window.FEATURE_NO_TITLE)
19     setCancelable(false)
20     setContentView(R.layout.dialog_form)
21
22     if (formLayout.parent != null) {
23         (formLayout.parent as ViewGroup).removeView(formLayout)
24     }
25
26     tv_title.text = title
27     form_container.addView(formLayout)
28     btn_cancel.setOnClickListener { dismiss() }
29     btn_save.setOnClickListener { it: View? -> Unit {
30         saveAction()
31         dismiss()
32     }}
33 }
34
35
36 }
```

16. Dan membuat layout untuk **dialog\_form** yang menjadi tampilan pada form dialog.

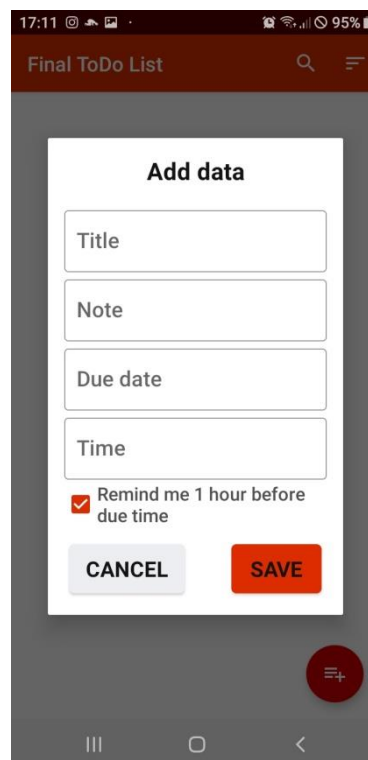
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     android:layout_width="match_parent"
5     android:layout_height="wrap_content"
6     xmlns:tools="http://schemas.android.com/tools"
7     android:background="#fff"
8     android:orientation="vertical"
9     android:padding="16dp">
10
11     <TextView
12         android:id="@+id/tv_title"
13         android:layout_width="match_parent"
14         android:layout_height="wrap_content"
15         android:layout_marginBottom="16dp"
16         android:gravity="center"
17         tools:text="Title"
18         android:textColor="@color/colorDarkGray"
19         android:textSize="18sp"
20         android:textStyle="bold" />
21
22     <ScrollView
23         android:id="@+id/form_container"
24         android:layout_width="match_parent"
25         android:layout_height="wrap_content"
26         android:orientation="vertical"
27         android:layout_marginBottom="10dp" />
28
29     <RelativeLayout
```

17. Membuat layout untuk tampilan **fragment\_todo.xml** untuk membuat fragmen-fragmen yang ingin diisi.

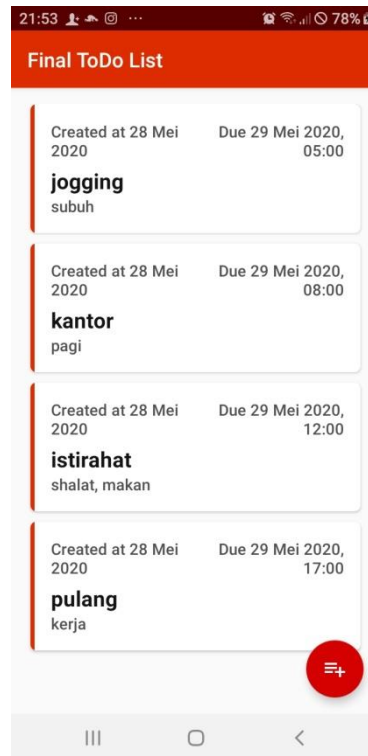


18. Kemudian run untuk memastikan apakah aplikasi sudah dapat melakukan fungsi CRUD, seperti dibawah ini:

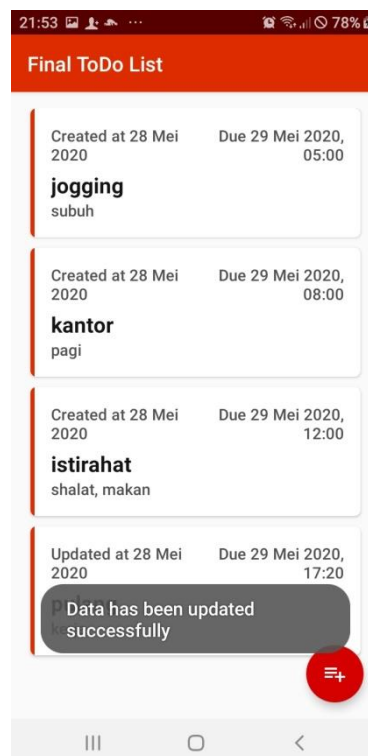
- **Create**



- Read

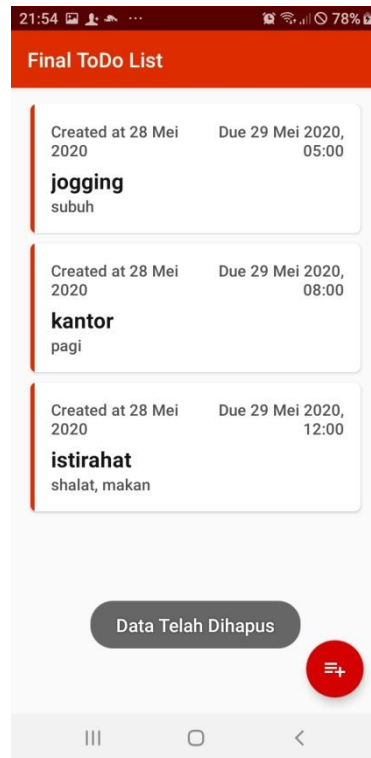


- Update

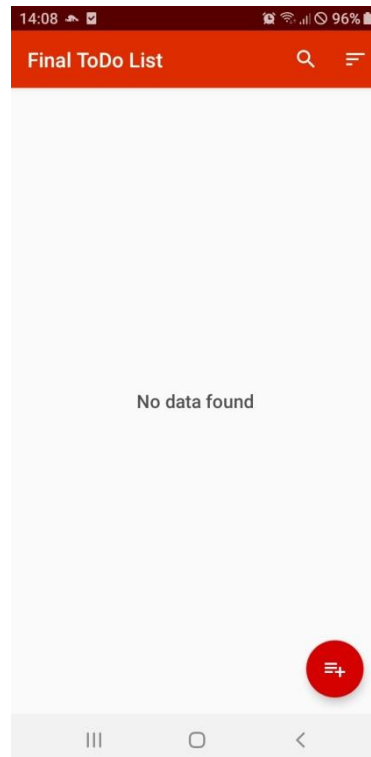




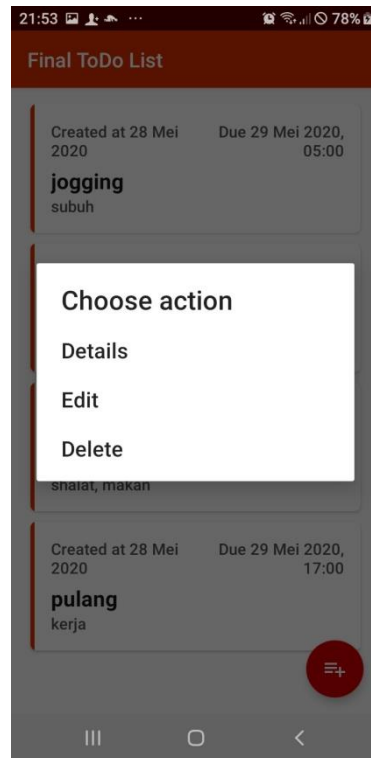
- **Delete**



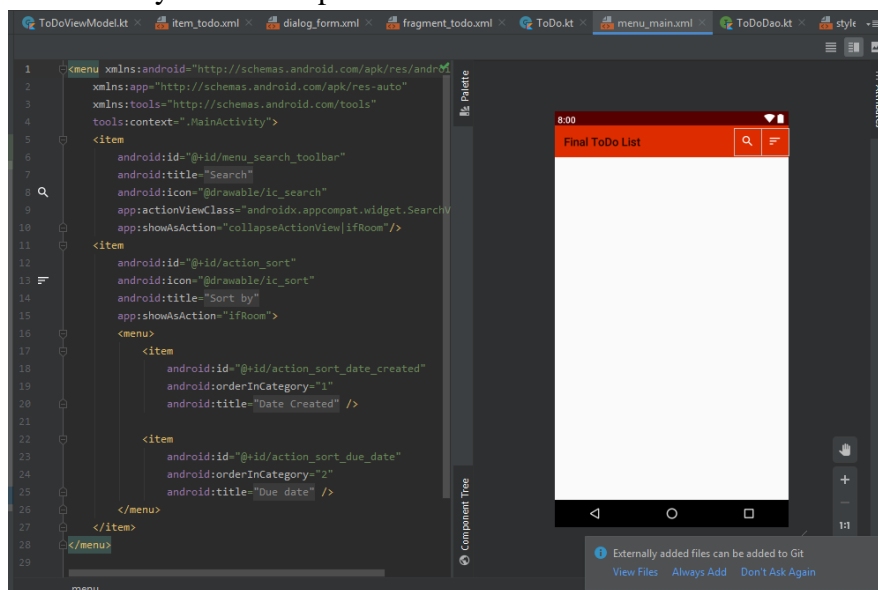
- **Empty**



## - Action

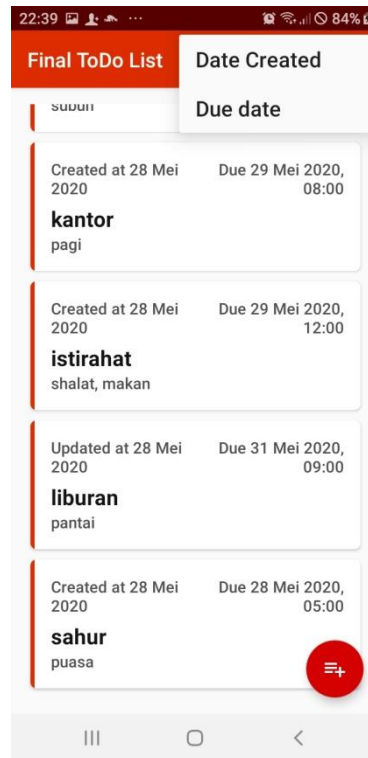


19. Setelah CRUD telah berfungsi sebagaimana mestinya, maka git push kedua akan dilakukan yang menandakan dua fitur telah selesai.
20. Membuat fitur sort yang dapat mengurut sebuah aktivitas/kegiatan yang telah ingin dilakukan.
21. Sebelum itu, buat beberapa file untuk men-sorting aktivitasnya. Adapun syntax nya terdapat pada file **MainActivity.kt**, **ToDoAdapter.kt**.
22. Kemudian, membuat menu dan icon sort nya pada file **ic\_sort.xml**, **menu\_main.xml**. Dan hasil nya akan berupa:

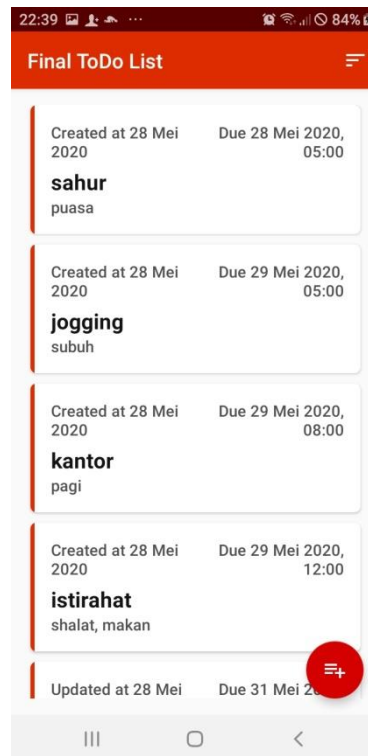


23. Setelah itu, maka run kembali untuk memastikan bahwa fitur sorting telah berfungsi, sebagai berikut:

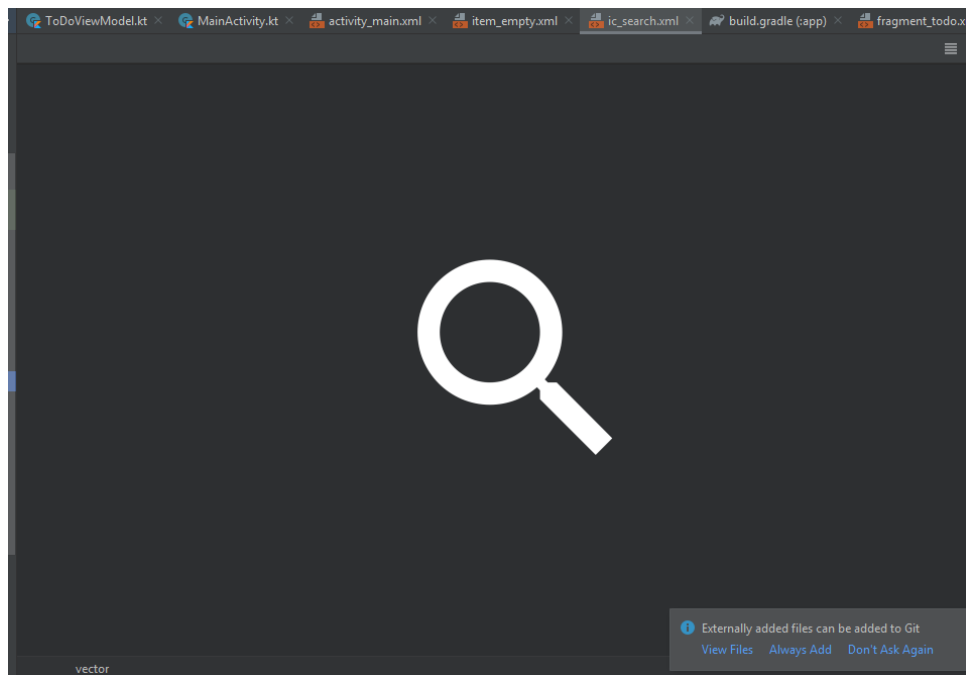
- **Menu Sort**



- **Hasil Sort**

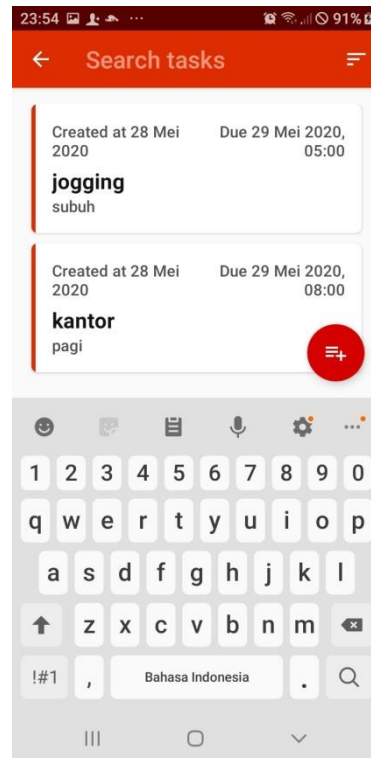


24. Hasil sorting yang dilakukan telah berfungsi mengikuti sort yang telah dipilih, yang awalnya urutan dimulai dari “Data Created” kemudian berubah jadi “Due date” yang men-sorting kegiatan dimulai dari batas tanggal yang terdekat.
25. Setelah fitur sorting telah berfungsi sebagaimana mestinya, maka git push ketiga akan dilakukan yang menandakan tiga fitur telah selesai.
26. Membuat fitur search yang dapat memfilter atau mencari aktivitas/kegiatan yang bersangkutan.
27. Sebelum itu, buat beberapa file untuk men-sorting aktivitasnya. Adapun syntax nya terdapat pada file **MainActivity.kt**, **ToDoAdapter.kt**.
28. Kemudian, membuat menu dan icon search nya pada file **ic\_search.xml**, **menu\_main.xml**. Dan hasil nya akan berupa:

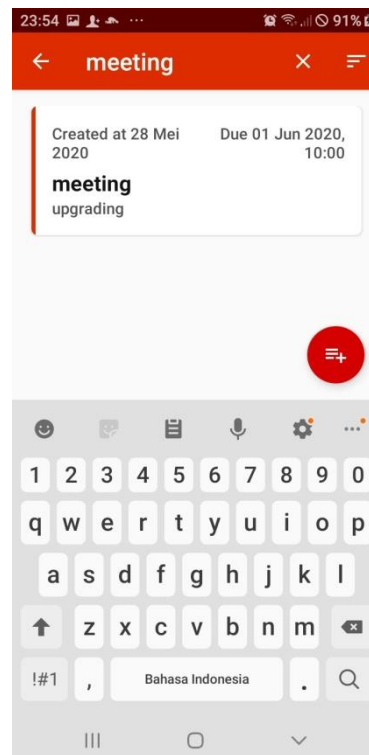


29. Setelah itu, maka run kembali untuk memastikan bahwa fitur searching telah berfungsi, sebagai berikut:

- **Search**



- **Hasil Search**



30. Hasil searching yang dilakukan telah berfungsi mengikuti sesuai perintah yang diinginkan.

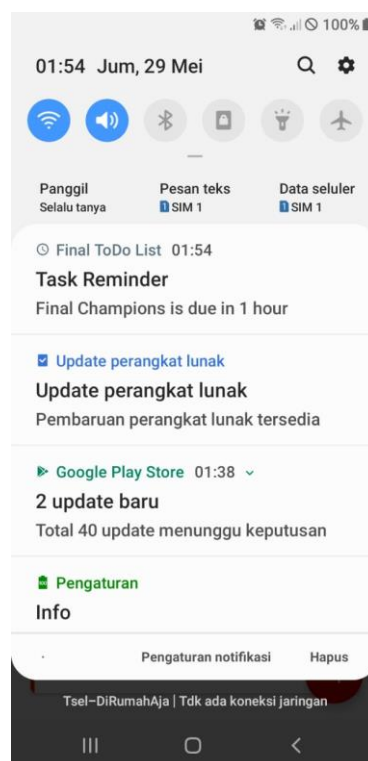
31. Setelah fitur searching telah berfungsi sebagaimana mestinya, maka git push keempat akan dilakukan yang menandakan empat fitur telah selesai.
32. Terakhir, fitur notification yang dapat menjadi pengingat 1 jam sebelum aktivitas akan dimulai.
33. Sebelum itu, buat beberapa file untuk men-sorting aktivitasnya. Adapun syntax nya terdapat pada file **AndroidManifest.xml**, **MainActivity.kt**, **AlarmReceiver.kt**, **ic\_access\_time.xml**.

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3  package="id.ac.unhas.finaltodolist">
4
5      <uses-permission android:name="android.permission.WAKE_LOCK" />
6      <uses-permission android:name="android.permission.VIBRATE" />
7
8      <application
9          android:allowBackup="true"
10         android:icon="@mipmap/ic_launcher"
11         android:label="Final To Do List"
12         android:roundIcon="@mipmap/ic_launcher_round"
13         android:supportRtl="true"
14         android:theme="@style/AppTheme">
15         <activity
16             android:name=".MainActivity"
17             android:label="Final To Do List">
18             <intent-filter>
19                 <action android:name="android.intent.action.MAIN" />
20                 <category android:name="android.intent.category.LAUNCHER" />
21             </intent-filter>
22         </activity>
23         <receiver
24             android:name=".util.AlarmReceiver"
25             android:enabled="true"
26             android:exported="true" />
27     </application>

```

34. Setelah itu, maka run kembali untuk memastikan bahwa fitur notification telah berfungsi, sebagai berikut:



35. Alarm tersebut akan menjadi pengingat bahwa kegiatan/aktivitas akan dimulai dalam 1 jam ke depan.
36. Setelah fitur notification telah berfungsi sebagaimana mestinya, maka git push kelima akan dilakukan yang menandakan lima fitur telah selesai.
37. Setelah semua fitur telah selesai, bentuk dari aplikasinya sebagai berikut:

