ILT Sesi 3 Android

Aturan Instructor/Expert Led Session

Fokus Penuh

Always-on Camera

Raise hand atau Chat jika ingin bertanya

Mute jika sedang tidak berbicara

Porsi Skor Penilaian (Tech)

Jenis Aktivitas	Porsi Skor
Rata-rata Kuis pada ILT	20%
Aktivitas selama ILT (bertanya, menjawab, membantu diskusi)	10%
Aktivitas Forum Diskusi*	10%
Rata-rata Exam/Submission Project	60%

*Jika ditemukan peserta spamming pada forum diskusi, maka nilai terbaik yang bisa didapat adalah 60 (dari skala 100)

dicoding Kampus Merdeka INDONESIA JAYA

Materi/Review

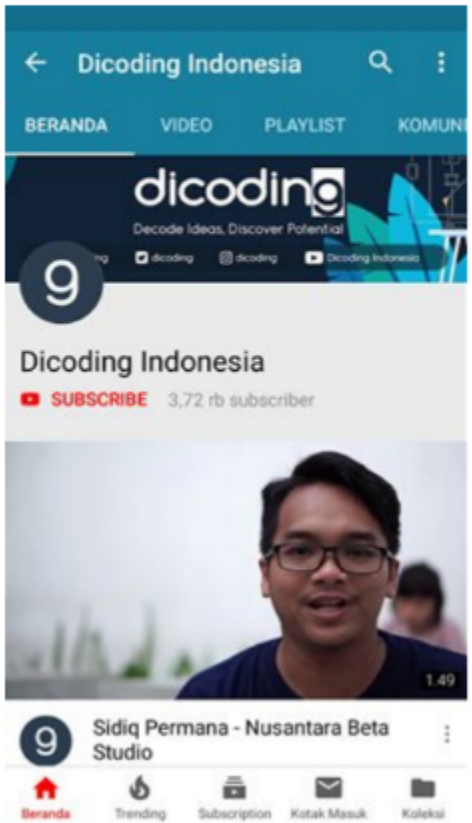
Materi yang Telah Dipelajari.

- **Belajar Fundamental Aplikasi Android**
  - Fundamental
  - Submission 1
  - Navigation
  - Background Process

Navigation

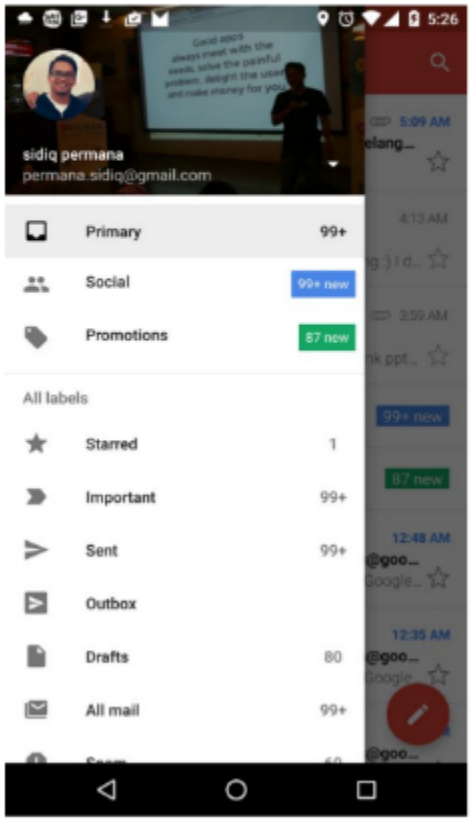
Jenis-Jenis Navigasi

Bottom Navigation View



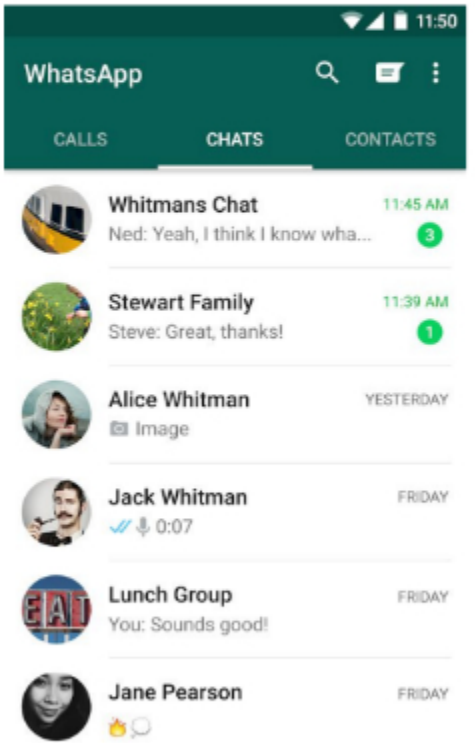
1

Navigation Drawer



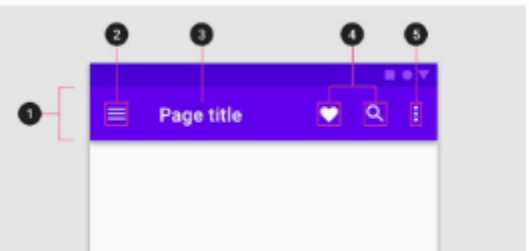
2

Tab Layout



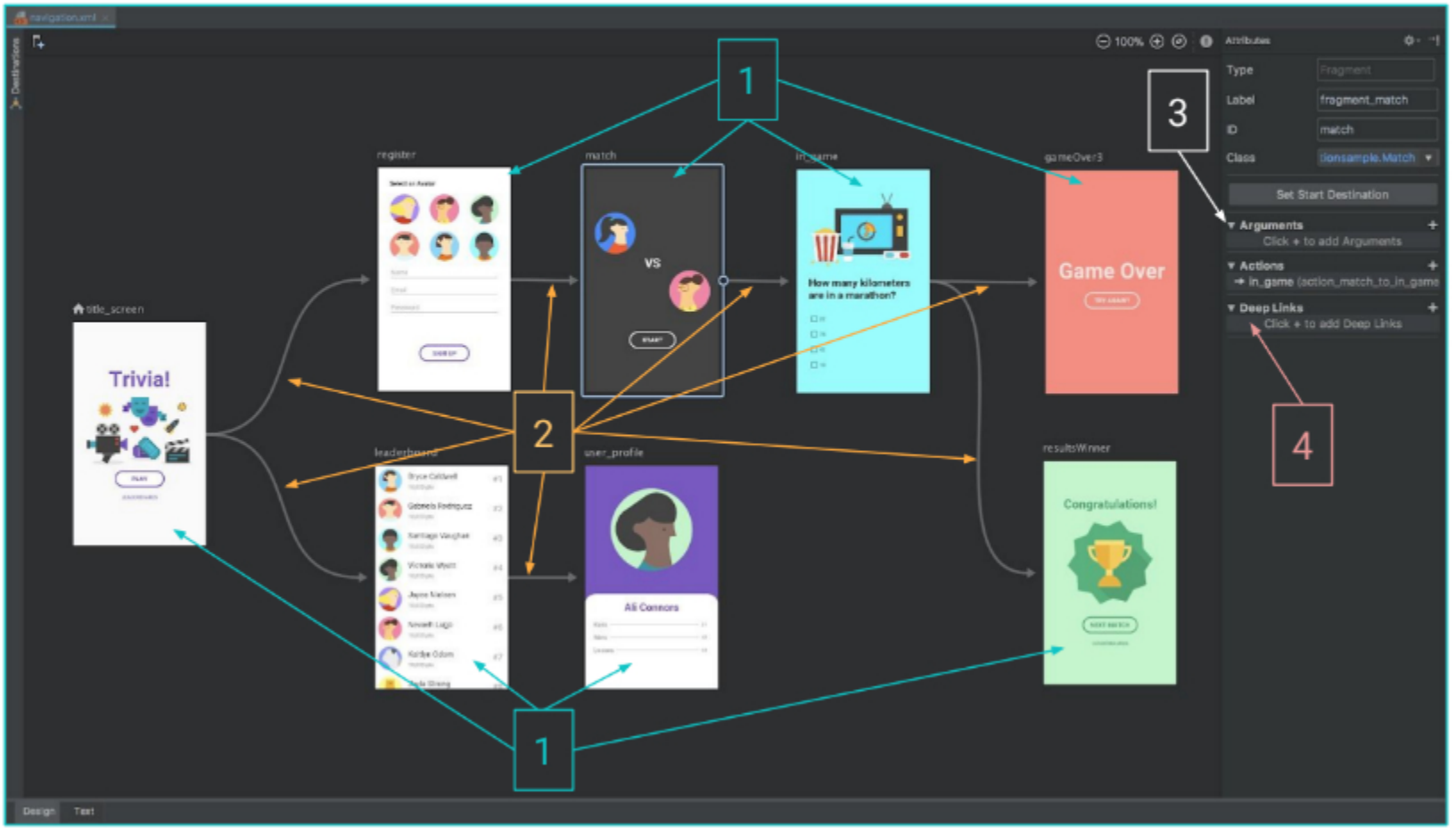
3

Action Bar



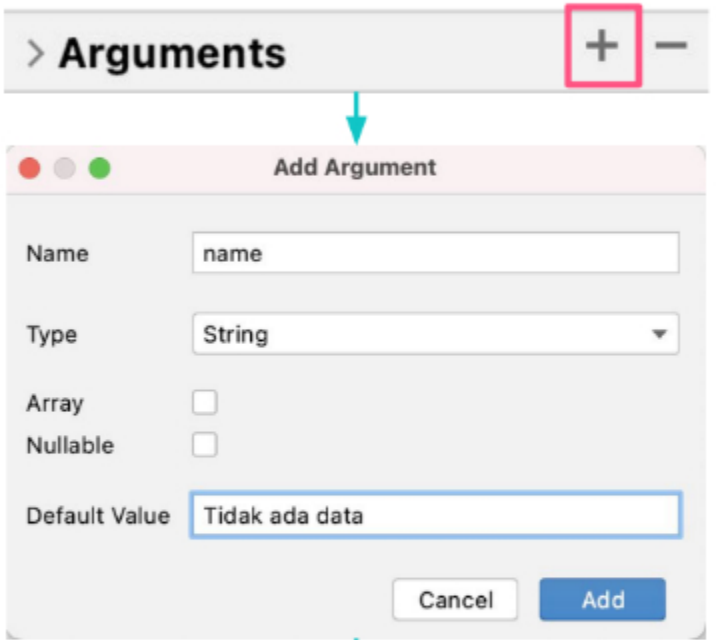
4

Navigation Component



1. Destination
2. Action
3. Argument
4. DeepLink

Mengirim Data pada Navigation Component




```
plugins {
  ...
  id "androidx.navigation.safeargs"
}
```

```
dependencies {
  ...
  classpath
"androidx.navigation:navigation-safe-args-gradle-plugin:2.3.5"
}
```

```
textView.setOnClickListener { view ->
  val toDetail = DashboardFragmentDirections
    .actionNavigationDashboardToNavigationNotifications()
  toDetail.name = "Habibi"
  view.findNavController().navigate(toDetail)
}
```

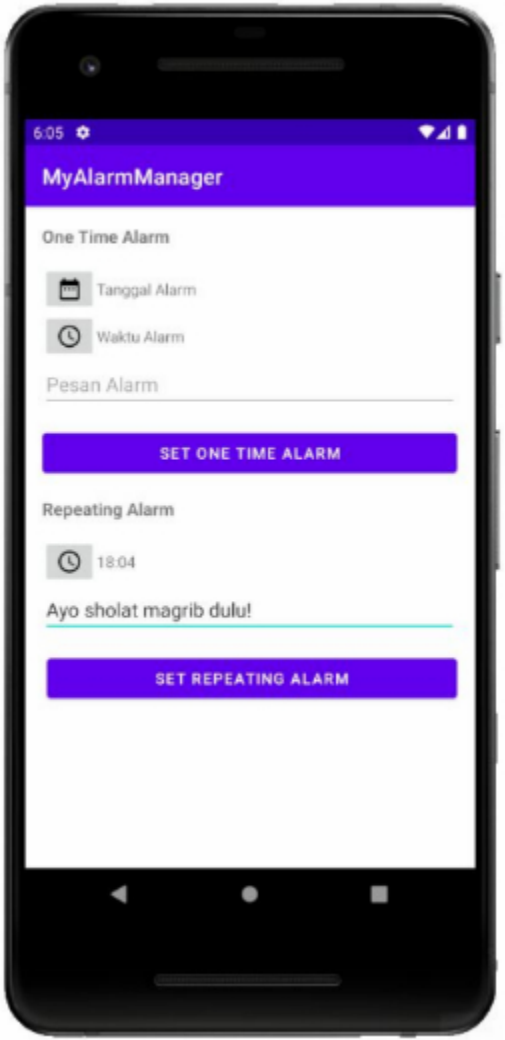
```
val dataName = NotificationsFragmentArgs
  .fromBundle(arguments as Bundle)
  .name
```

dicoding Kampus Merdeka INDONESIA JAYA

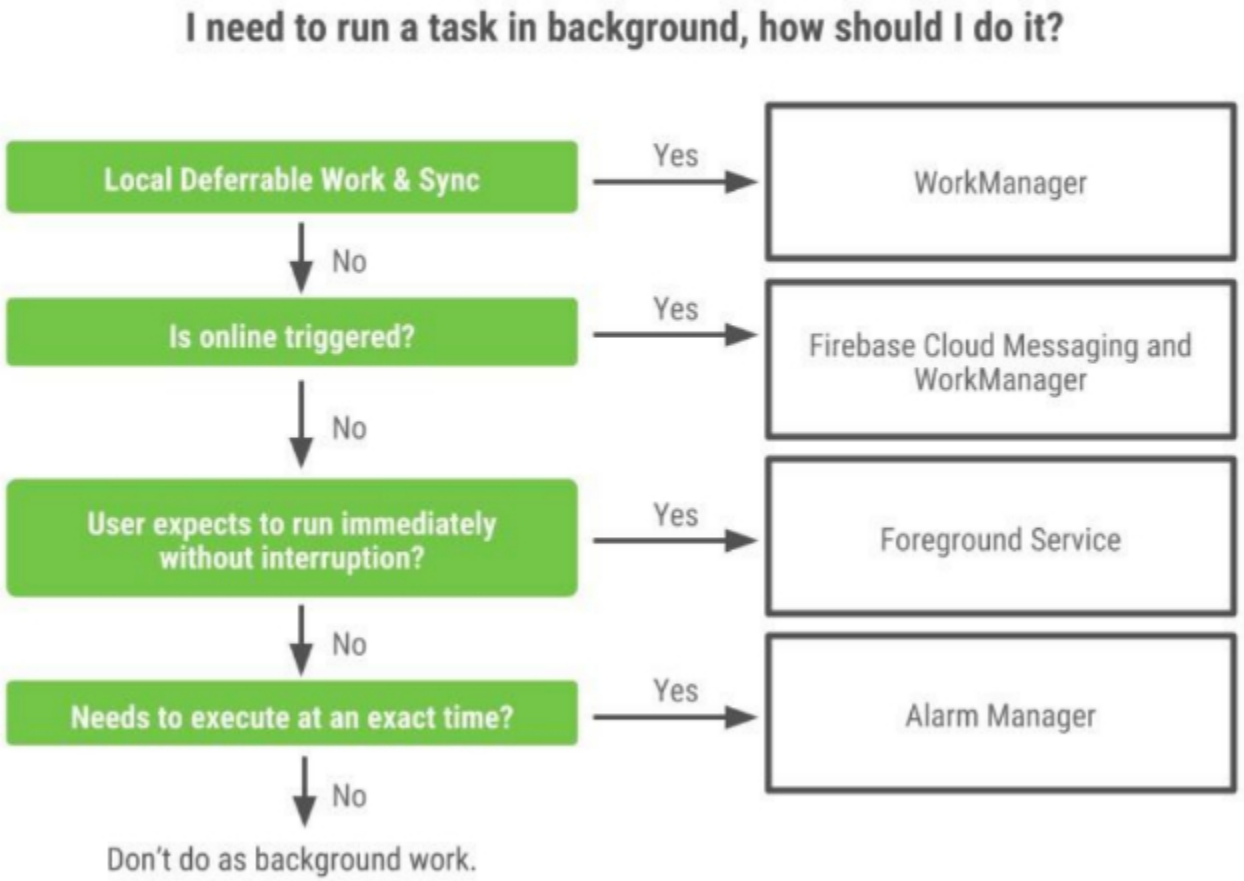
Scheduler

Alarm Manager

- Menjalankan object intent berdasarkan waktu dan interval yang ditentukan.
- Bisa bekerja dengan baik bersama broadcast receiver untuk menjalankan komponen lain seperti service untuk melakukan operasi tertentu.
- Berjalan di luar daur hidup aplikasi induknya.
- Meminimalkan penggunaan resource dan menghindari penggunaan timer dan background service yang berkepanjangan.

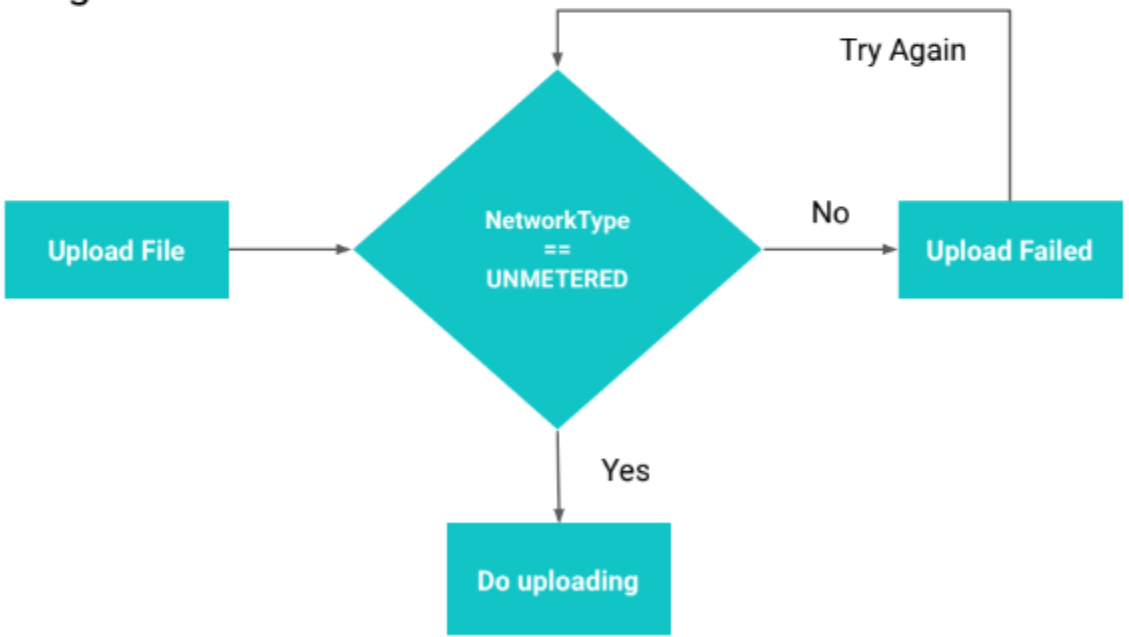


Work Manager



Studi Kasus WorkManager

Dengan menggunakan WorkManager, kita bisa mengatur *requirement* dari suatu task untuk berjalan pada kondisi tertentu. Semisal pada kasus ini adalah upload file pada Google Drive.



Networking

dicoding Kampus Merdeka INDONESIA JAYA

Retrofit - Square Open Source

Retrofit merupakan salah satu library http client atau library yang biasa digunakan untuk berkomunikasi dengan api. Baik itu **POST**, **GET**, **DELETE**, **PUT** atau metode lainnya.

Keunggulan retrofit dibanding library lainnya adalah penggunaannya yang cukup **mudah**. Selain itu, retrofit juga sudah bisa digunakan dengan Coroutine, LiveData, dan RxJava.

dicoding Kampus Merdeka INDONESIA JAYA

Penggunaan Retrofit secara Default

```
interface ApiInterface {
    @GET("api/recipes") // End-Point dari API yang diminta menggunakan method GET.
    fun getRecipes(): Call<Recipes> // Menentukan data class yang cocok dengan response API.
}

private fun getApiService(): ApiInterface {
    val retrofit: Retrofit = Retrofit.Builder() // Membuat builder untuk Retrofit.
        .baseUrl("https://masak-apa.tomorisakura.vercel.app") // URL API yang akan dipanggil.
        .addConverterFactory(GsonConverterFactory.create()).build() // Mengonversi JSON menjadi data class.
    return retrofit.create(ApiInterface::class.java) // Menghubungkan Retrofit dengan ApiInterface.
}

private fun getRecipes() {
    getApiService().getRecipes().enqueue(object : Callback<Recipes> { // Memanggil Method getRecipes().
        override fun onResponse(call: Call<Recipes>, response: Response<Recipes>) {
            response.body()?.let {
                showRecipes(it.results) // Ketika data-nya ada, bisa ditampilkan ke RecyclerView.
            }
        }
        override fun onFailure(call: Call<Recipes>, t: Throwable) {
            println("data gagal didapatkan, error $t") // Ketika gagal, akan ditampilkan ke console.
        }
    }
}
```

dicoding Kampus Merdeka INDONESIA JAYA

Penggunaan Retrofit dengan Coroutine

```
interface ApiInterface {
    @GET("api/recipes") // End-Point dari API yang diminta menggunakan method GET.
    suspend fun getRecipes(): Recipes // Menggunakan suspend agar dapat dijalankan dengan Coroutine.
}

private fun getApiService(): ApiInterface {
    val retrofit: Retrofit = Retrofit.Builder() // Membuat builder untuk Retrofit.
        .baseUrl("https://masak-apa.tomorisakura.vercel.app") // URL API yang akan dipanggil.
        .addConverterFactory(GsonConverterFactory.create()).build() // Mengonversi JSON menjadi data class.
    return retrofit.create(ApiInterface::class.java) // Menghubungkan Retrofit dengan ApiInterface.
}

private suspend fun getRecipes(){
    val response = getApiService().getRecipes() // Memanggil Method getRecipes().
    runOnUiThread { showRecipes(response.results) } // Menampilkan data pada main thread.
}

override fun onCreate(savedInstanceState: Bundle?) {
    CoroutineScope(Dispatchers.IO).launch { // Membuat thread baru
        getRecipes() // Memanggil fungsi getRecipes pada thread yang sudah dibuat
    }
}
```

dicoding Kampus Merdeka INDONESIA JAYA

Sesi Sharing

dicoding Kampus Merdeka INDONESIA JAYA

Ujian

Sesi Diskusi


dicoding


Kampus
Merdeka
INDONESIA JAYA