

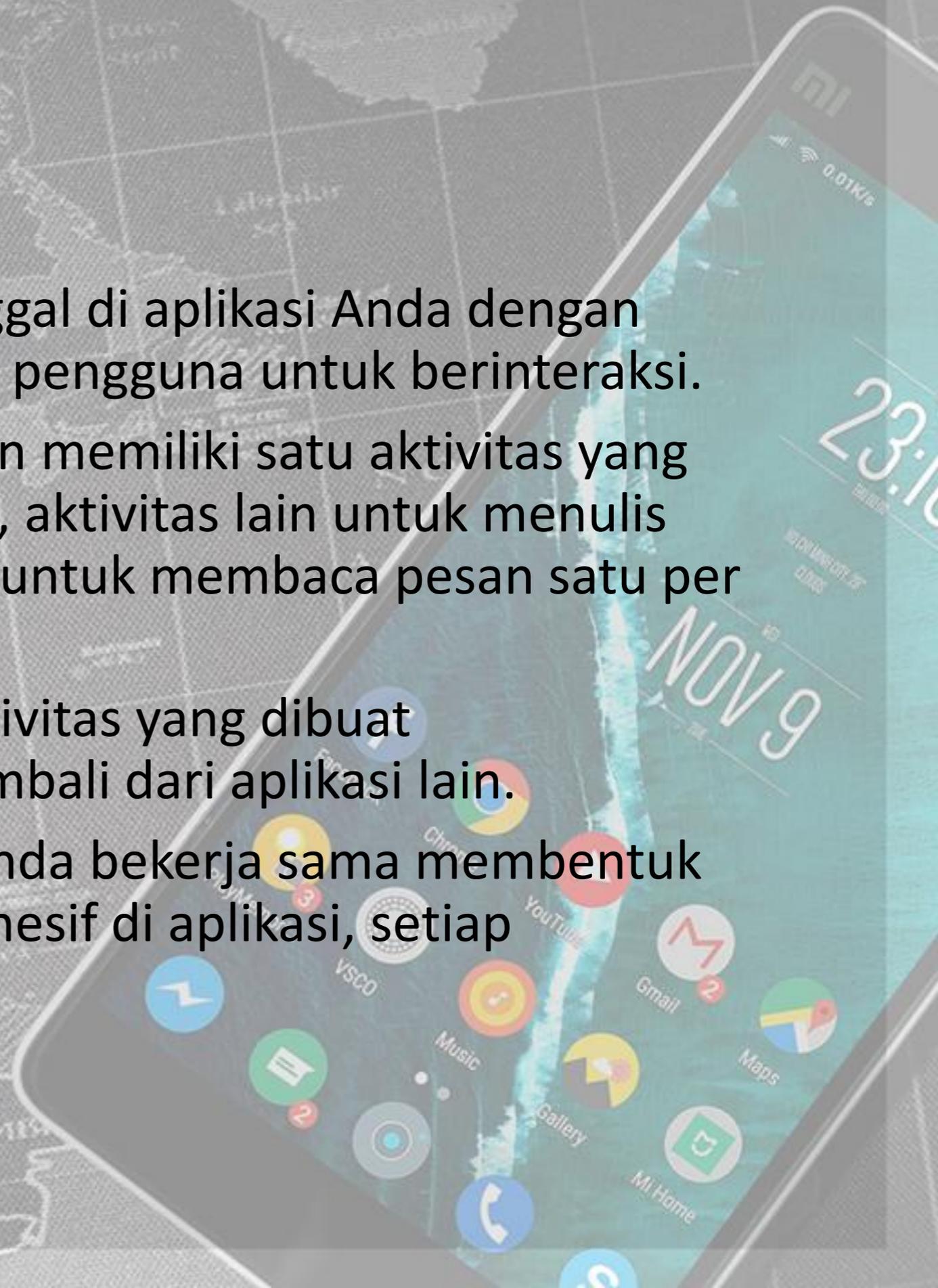
# Android Activity

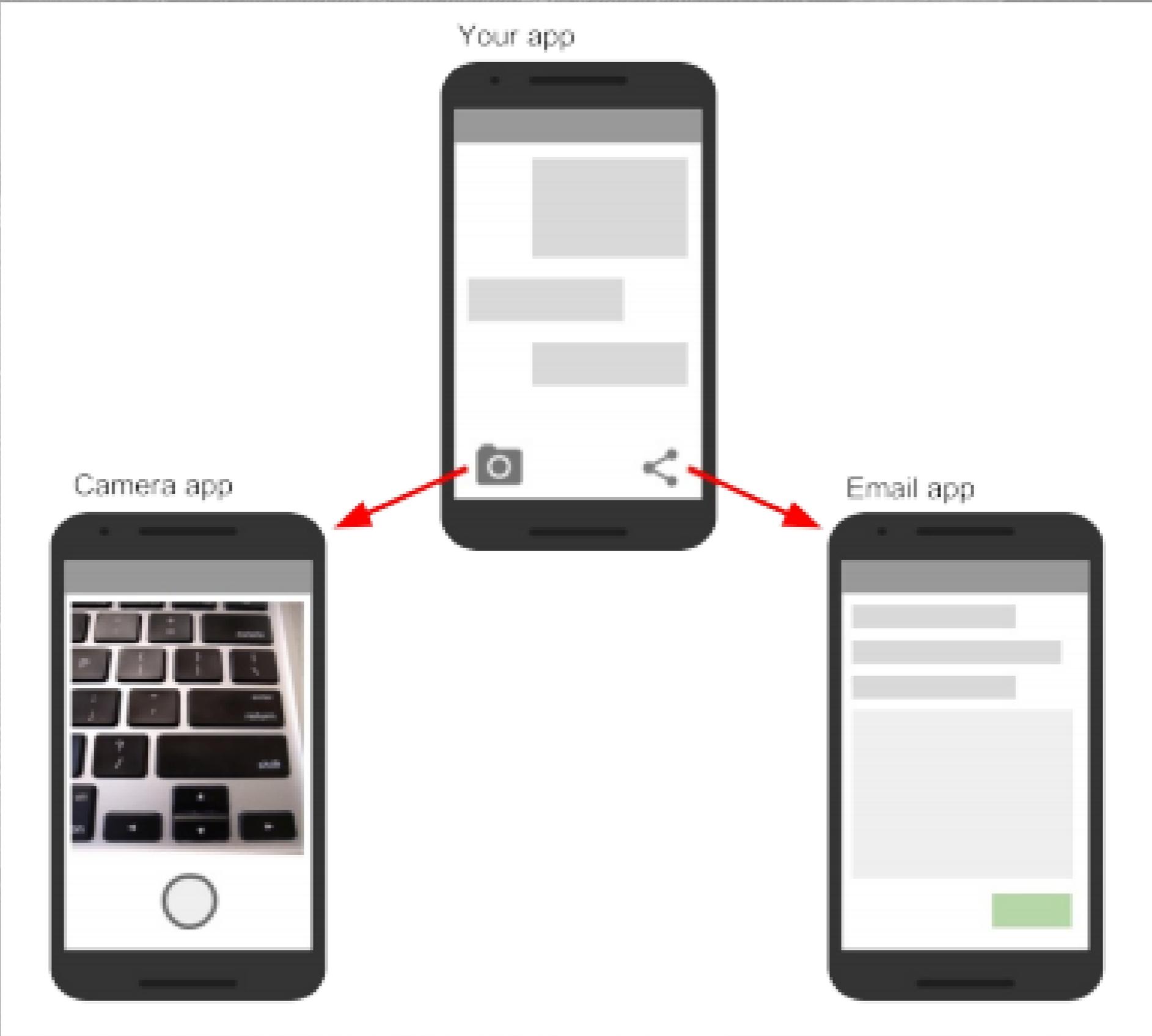
Muhammad A. Safi'ie, S.Si, M.Kom  
D3 Teknik Informatika  
Universitas Sebelas Maret



# Activity

- Aktivitas menyatakan layar tunggal di aplikasi Anda dengan antarmuka yang bisa digunakan pengguna untuk berinteraksi.
- Misalnya, aplikasi email mungkin memiliki satu aktivitas yang menampilkan daftar email baru, aktivitas lain untuk menulis email, dan aktivitas lainnya lagi untuk membaca pesan satu per satu.
- Aplikasi Anda adalah koleksi aktivitas yang dibuat sendiri atau yang digunakan kembali dari aplikasi lain.
- Meskipun aktivitas di aplikasi Anda bekerja sama membentuk pengalaman pengguna yang kohesif di aplikasi, setiap aplikasi tidak saling bergantung





# Intent

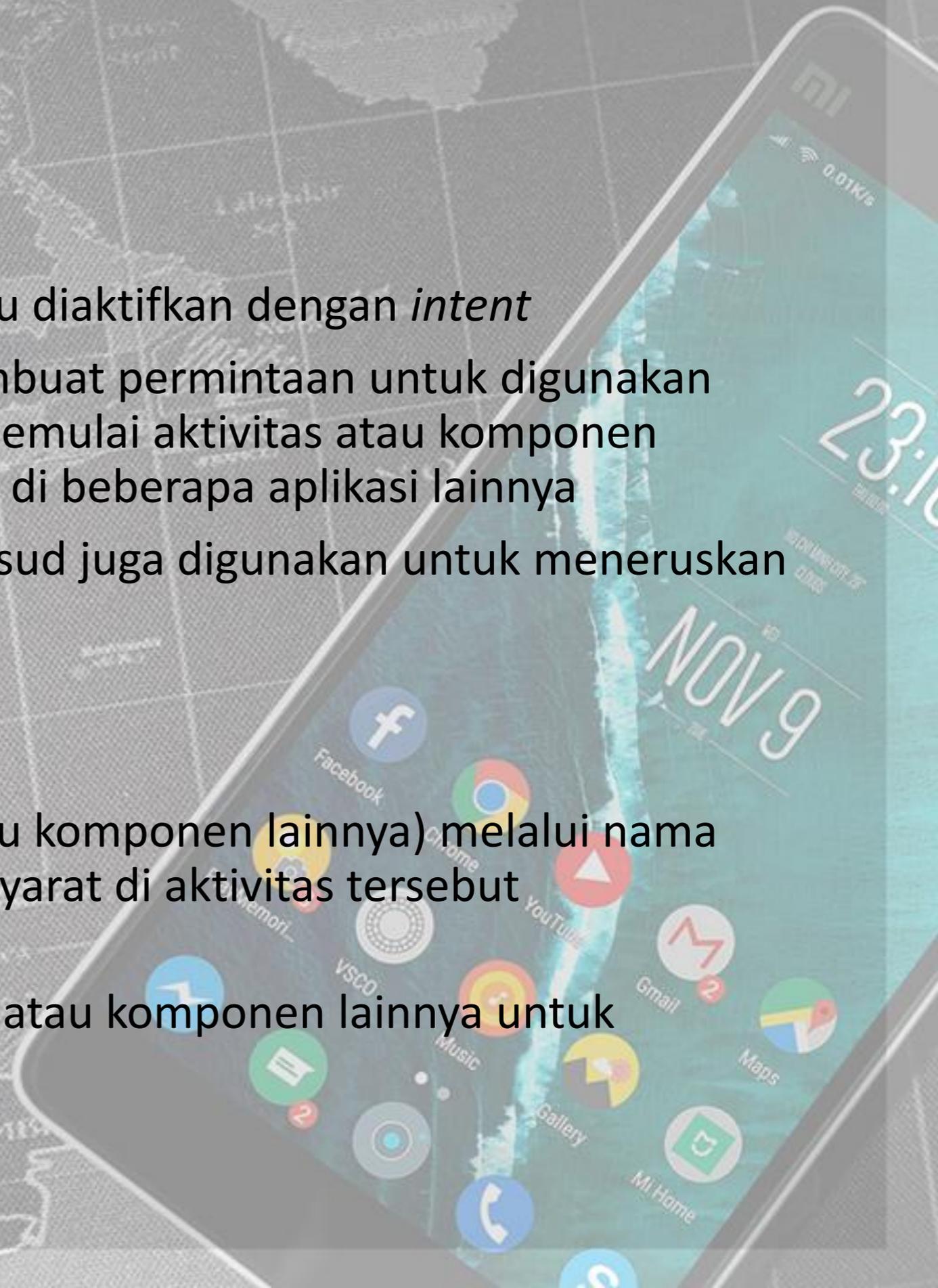
- Semua aktivitas Android dimulai atau diaktifkan dengan *intent*
- *Intent* adalah objek pesan yang membuat permintaan untuk digunakan pada waktu proses Android untuk memulai aktivitas atau komponen aplikasi lainnya di aplikasi Anda atau di beberapa aplikasi lainnya
- Selain untuk memulai aktivitas, maksud juga digunakan untuk meneruskan data di antara aktivitas.
- Tipe Intent

## Intent Eksplisit

menetapkan aktivitas penerima (atau komponen lainnya) melalui nama kelas yang benar-benar memenuhi syarat di aktivitas tersebut

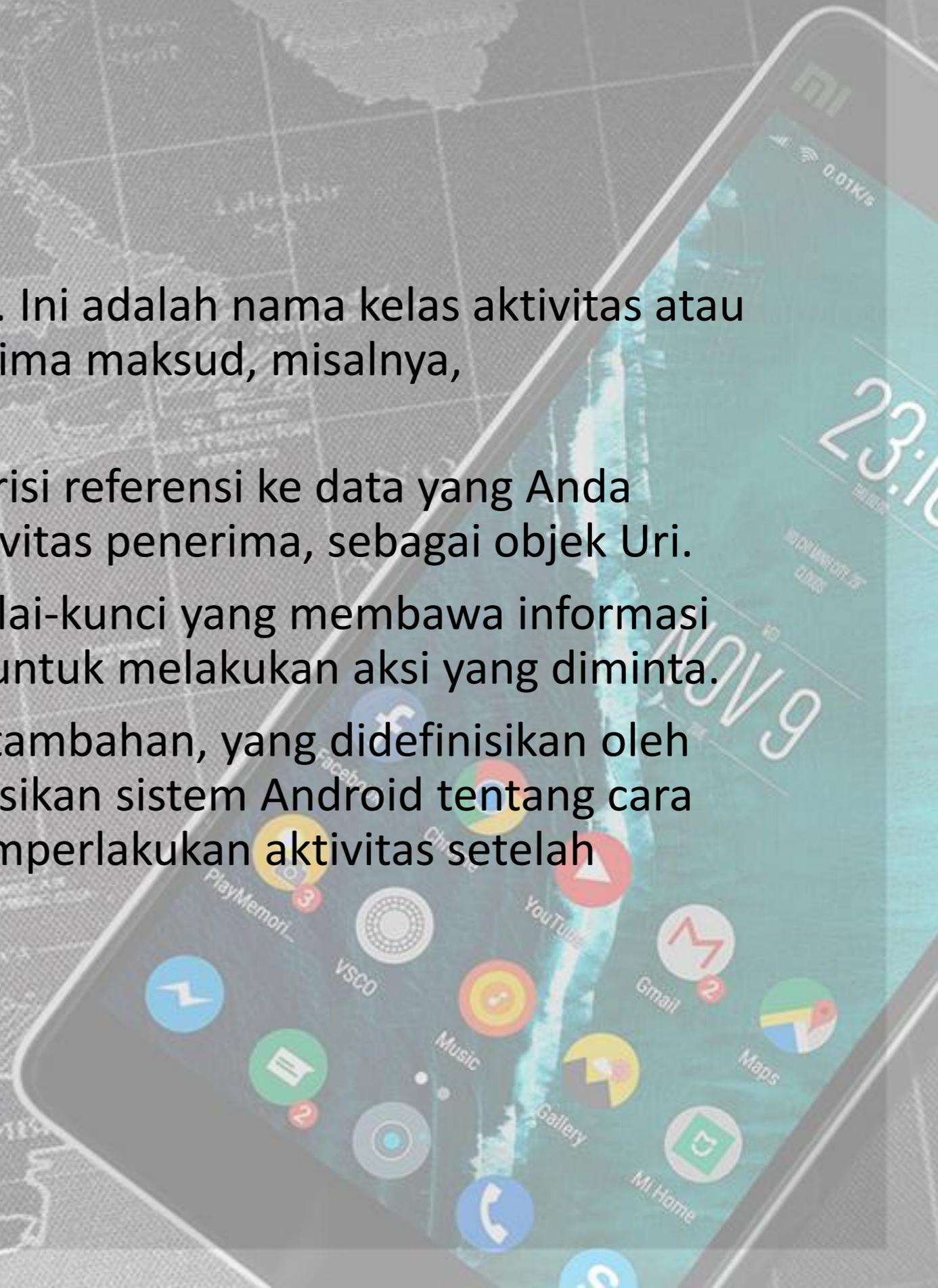
## Intent Implisit

tidak menetapkan aktivitas tertentu atau komponen lainnya untuk menerima *intent*.



# Objek dan bidang Intent

- *Activity Class* (untuk *intent* eksplisit). Ini adalah nama kelas aktivitas atau komponen lainnya yang akan menerima maksud, misalnya, `com.example.SampleActivity.class`
- *Intent Data*. Bidang data maksud berisi referensi ke data yang Anda inginkan untuk mengoperasikan aktivitas penerima, sebagai objek Uri.
- *Intent Ekstra*. Ini adalah pasangan nilai-kunci yang membawa informasi yang diperlukan aktivitas penerima untuk melakukan aksi yang diminta.
- *Intent Flag*. Ini adalah bit metadata tambahan, yang didefinisikan oleh kelas Intent. Flag dapat menginstruksikan sistem Android tentang cara meluncurkan aktivitas atau cara memperlakukan aktivitas setelah diluncurkan



# Memulai aktivitas dengan maksud eksplisit

- Misalnya, jika Anda ingin memulai ShowMessageActivity untuk menampilkan pesan tertentu di aplikasi email, gunakan kode seperti ini

```
Intent messageIntent = new Intent(this, ShowMessageActivity.class);
startActivity(messageIntent);
```

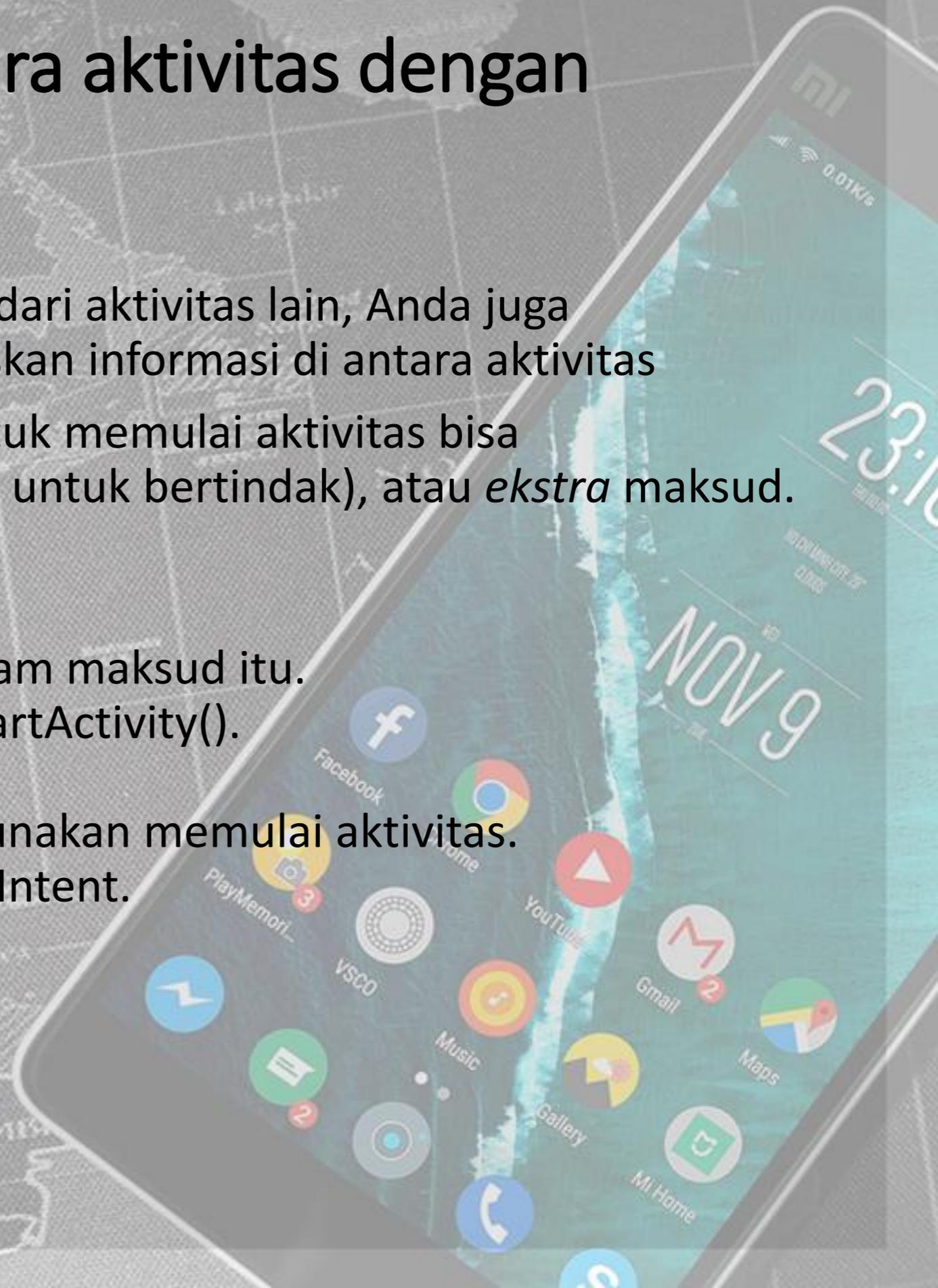
- Konstruktor Intent menggunakan dua argumen untuk maksud eksplisit. Konteks aplikasi. Dalam contoh ini, kelas aktivitas menyediakan materi (di sini, this ).  
Komponen tertentu untuk dimulai ( ShowMessageActivity. class )
- Menutup aktivitas

```
public void closeActivity (View view) {
    finish();
}
```



# Meneruskan data di antara aktivitas dengan maksud

- Selain untuk memulai satu aktivitas dari aktivitas lain, Anda juga menggunakan *intent* untuk meneruskan informasi di antara aktivitas
- Objek *intent* yang Anda gunakan untuk memulai aktivitas bisa menyertakan *data intent* (URI objek untuk bertindak), atau *ekstra* maksud.
- Di aktivitas (pengirim) pertama:
  1. Buat objek Intent.
  2. Masukkan data atau ekstra ke dalam maksud itu.
  3. Mulailah aktivitas baru dengan `startActivity()`.Di aktivitas (penerima) kedua:
  1. Dapatkan objek maksud yang digunakan memulai aktivitas.
  2. Ambil data atau ekstra dari objek Intent.



# Waktu untuk menggunakan data maksud atau ekstra maksud

- *Intent* ekstra untuk meneruskan data di antara aktivitas.
- Ada sejumlah perbedaan utama antara data dan ekstra yang menentukan mana yang harus digunakan.
- *Intent Data* hanya bisa menyimpan satu bagian informasi
- Gunakan bidang maksud data:  
Bila Anda hanya memiliki satu bagian informasi yang perlu dikirim ke aktivitas yang telah dimulai.  
Bila informasi itu adalah lokasi data yang bisa dinyatakan dengan URI.
- *Ekstra* maksud adalah untuk data arbitrer lainnya yang ingin Anda teruskan ke aktivitas yang telah dimulai.
- Ekstra maksud disimpan di objek Bundle sebagai pasangan kunci dan nilai
- Gunakan ekstra maksud:  
Jika Anda ingin meneruskan lebih dari satu bagian informasi ke aktivitas yang telah dimulai.  
Jika informasi yang ingin Anda teruskan tidak bisa dinyatakan melalui URI.

# Tambahkan data ke *Intent*

- Untuk menambahkan data ke maksud eksplisit dari aktivitas pembuatnya, buat objek *intent* seperti berikut:

```
Intent messageIntent = new Intent(this, ShowMessageActivity.class);
```

- Gunakan metode setData() bersama objek Uri untuk menambahkan URI itu ke *intent*. Beberapa contoh penggunaan setData() bersama URI:

```
// A web page URL
messageIntent.setData(Uri.parse("http://www.google.com"));
// a Sample file URI
messageIntent.setData(Uri.fromFile(new File("/sdcard/sample.jpg")));
// A sample content: URI for your app's data model
messageIntent.setData(Uri.parse("content://mysample.provider/data"));
// Custom URI
messageIntent.setData(Uri.parse("custom:" + dataID + buttonId));
```

- Perlu diingat bahwa bidang data hanya bisa berisi URI tunggal

# Tambahkan ekstra ke maksud

- Untuk menambahkan ekstra maksud ke maksud eksplisit dari aktivitas pembuatnya:
  1. Tentukan kunci yang akan digunakan untuk informasi yang ingin dimasukkan ke dalam ekstra, atau definisikan sendiri. Setiap bagian informasi memerlukan kunci unik.
  2. Gunakan metode putExtra() untuk menambahkan pasangan kunci/nilai ke ekstra maksud. Secara opsional Anda bisa membuat objek Bundle, menambahkan data ke bundel, kemudian menambahkan bundel ke maksud
- Secara konvensional Anda mendefinisikan kunci ekstra maksud sebagai variabel-variabel statis dengan nama yang dimulai kata EXTRA\_

```
public final static String EXTRA_MESSAGE = "com.example.mysampleapp.MESSAGE";  
public final static String EXTRA_POSITION_X = "com.example.mysampleapp.X";  
public final static String EXTRA_POSITION_Y = "com.example.mysampleapp.Y";
```

- Gunakan metode `putExtra()` bersama kunci untuk memasukkan data ke dalam ekstra maksud. Kelas `Intent` mendefinisikan banyak metode `putExtra()` untuk jenis data yang berbeda:

```
messageIntent.putExtra(EXTRA_MESSAGE, "this is my message");  
messageIntent.putExtra(EXTRA_POSITION_X, 100);  
messageIntent.putExtra(EXTRA_POSITION_Y, 500);
```

- Atau, Anda bisa membuat bundel baru dan mengisinya dengan ekstra maksud. Bundel mendefinisikan banyak metode "put" untuk jenis data primitif yang berbeda serta objek yang mengimplementasikan antarmuka `Parcelable` Android atau `Serializable` Java.

```
Bundle extras = new Bundle();  
extras.putString(EXTRA_MESSAGE, "this is my message");  
extras.putInt(EXTRA_POSITION_X, 100);  
extras.putInt(EXTRA_POSITION_Y, 500);
```

# Ambil data dari maksud di aktivitas yang dimulai

- Untuk mengambil maksud yang digunakan untuk memulai aktivitas (atau komponen lain), gunakan metode getIntent():

```
Intent intent = getIntent();
```

- Gunakan getData() untuk mendapatkan URI dari maksud itu:

```
Uri locationUri = getData();
```

- Gunakan salah satu metode getExtra() untuk mengekstrak data ekstra dari objek maksud:

```
String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);  
int positionX = intent.getIntExtra(MainActivity.EXTRA_POSITION_X);  
int positionY = intent.getIntExtra(MainActivity.EXTRA_POSITION_Y);
```

- Atau Anda bisa mendapatkan seluruh bundel ekstra dari maksud dan mengekstrak nilai dengan beragam metode Bundle:

```
Bundle extras = intent.getExtras();  
String message = extras.getString(MainActivity.EXTRA_MESSAGE);
```

# Mendapatkan data kembali dari aktivitas

- Bila Anda memulai aktivitas bersama sebuah maksud, aktivitas pembuatnya akan dihentikan sementara, dan aktivitas baru tetap di layar hingga pengguna mengeklik tombol kembali, atau panggil metode finish() di handler klik atau fungsi lainnya yang mengakhiri keterlibatan pengguna pada aktivitas ini.
- Untuk meluncurkan aktivitas baru dan mendapatkan kembali hasilnya, lakukan langkah-langkah berikut di aktivitas pembuatnya:
  1. Sebagai ganti meluncurkan aktivitas bersama startActivity(), panggil startActivityForResult() bersama maksud dan kode permintaan.
  2. Buat maksud baru di aktivitas yang diluncurkan dan tambahkan data yang dikembalikan ke maksud itu.
  3. Implementasikan onActivityResult() di aktivitas pembuatnya untuk memproses data yang