

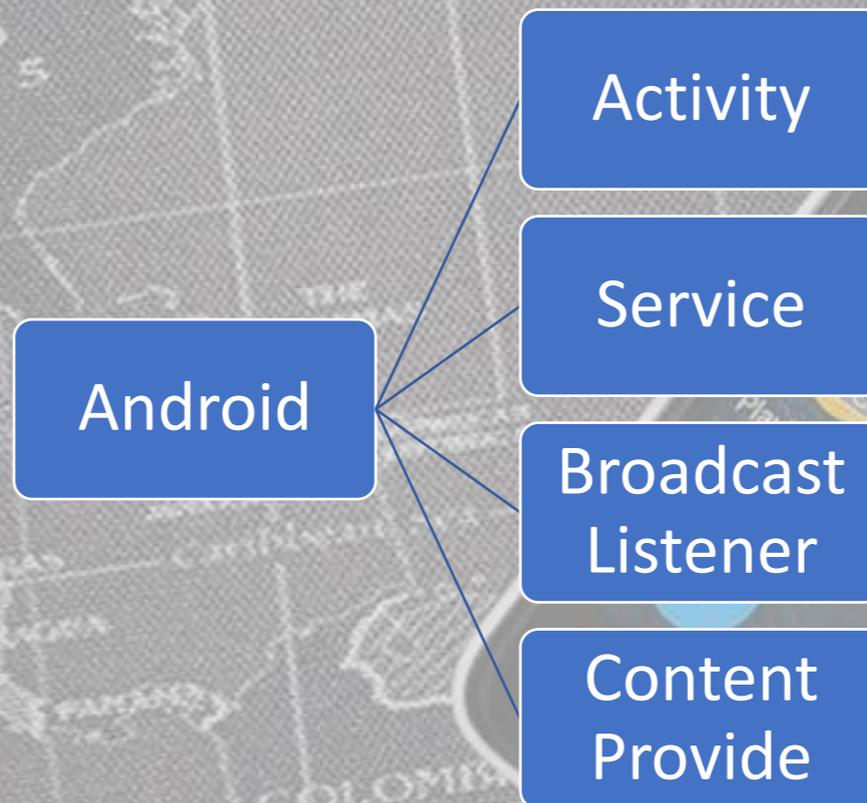
Anatomi Aplikasi Android

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



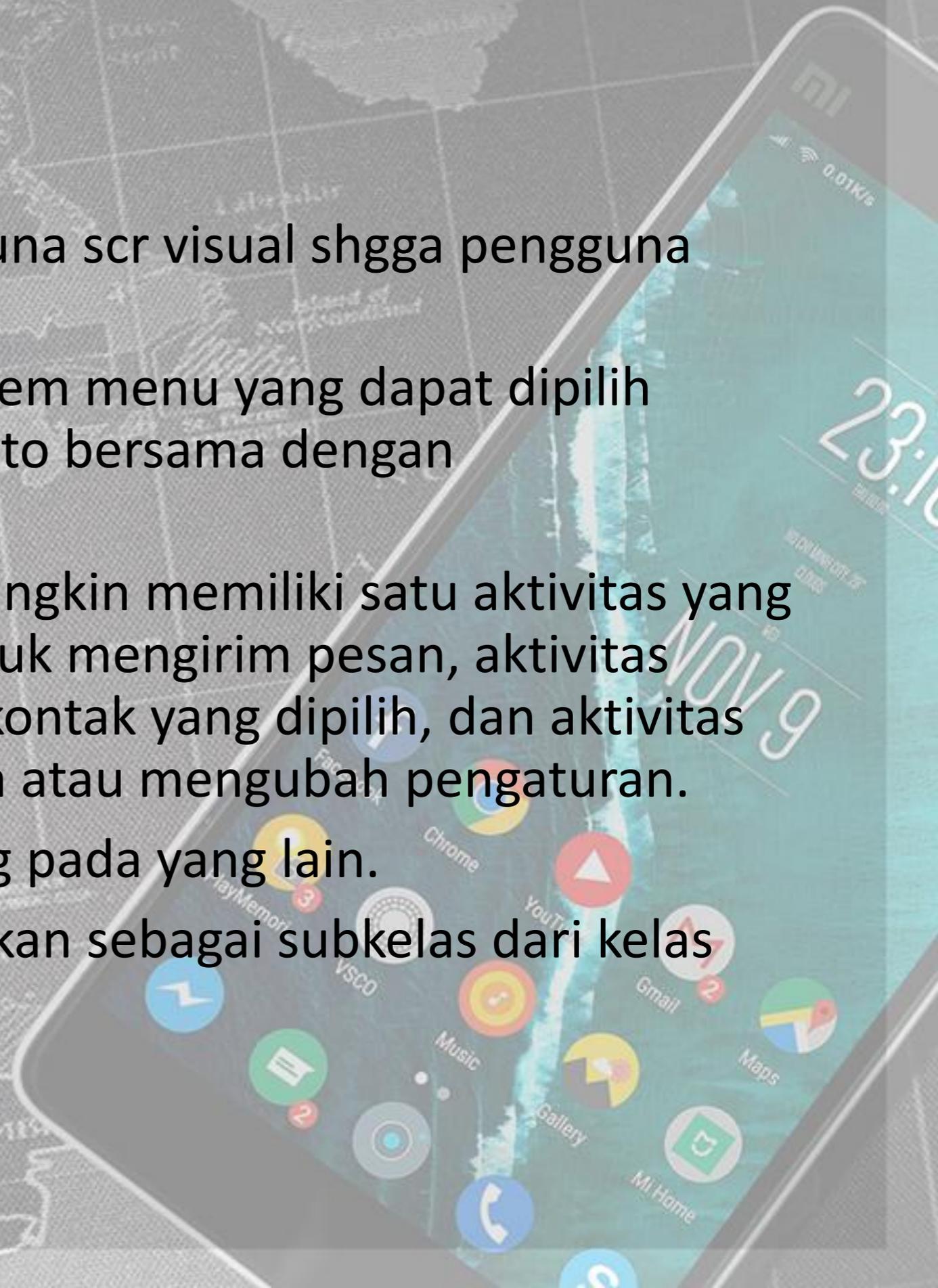
Tipe Komponen Aplikasi

- Ada empat macam tipe komponen aplikasi. Setiap tipe memiliki kegunaan tersendiri dan daur hidupnya sendiri yang mendefinisikan cara komponen dibuat dan dimusnahkan.



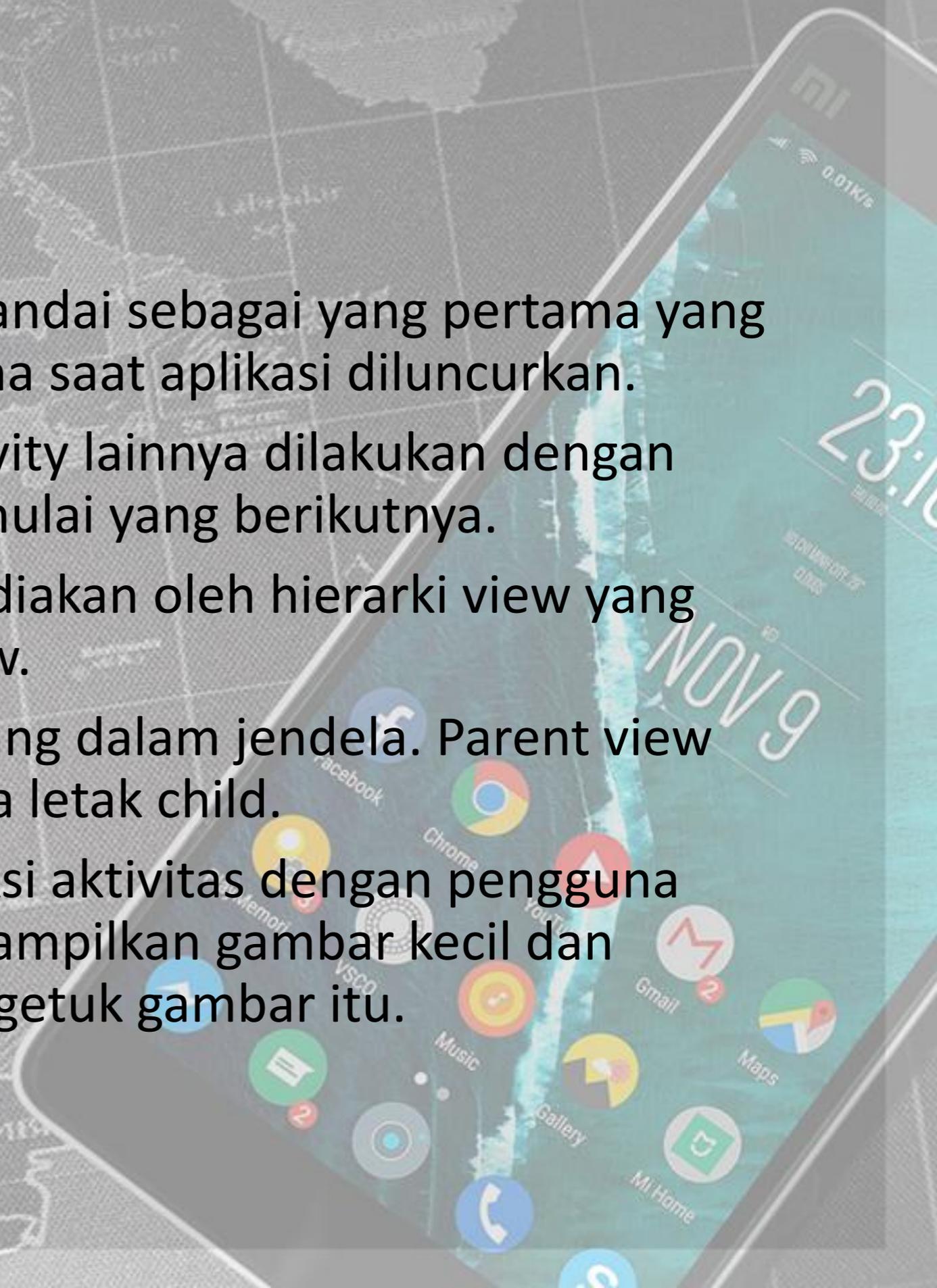
Activity

- Menampilan antarmuka pengguna scr visual shgga pengguna hanya fokus pada 1 saja.
- Contoh - Menampilkan daftar item menu yang dapat dipilih pengguna atau menampilkan foto bersama dengan keterangannya.
- Contoh - Aplikasi pesan teks mungkin memiliki satu aktivitas yang menunjukkan daftar kontak untuk mengirim pesan, aktivitas kedua untuk menulis pesan ke kontak yang dipilih, dan aktivitas lain untuk meninjau pesan lama atau mengubah pengaturan.
- Setiap aktivitas tidak tergantung pada yang lain.
- Masing-masing diimplementasikan sebagai subkelas dari kelas basis Kegiatan.



Activity...

- Biasanya, salah satu activity ditandai sebagai yang pertama yang harus disajikan kepada pengguna saat aplikasi diluncurkan.
- Pindah dari satu activity ke activity lainnya dilakukan dengan melakukan activity saat ini memulai yang berikutnya.
- Konten visual dari window disediakan oleh hierarki view yang diturunkan dari kelas dasar View.
- Setiap tampilan mengontrol ruang dalam jendela. Parent view mengandung dan mengatur tata letak child.
- Tampilan adalah tempat interaksi aktivitas dengan pengguna terjadi. Tampilan mungkin menampilkan gambar kecil dan merespons saat pengguna mengetuk gambar itu.



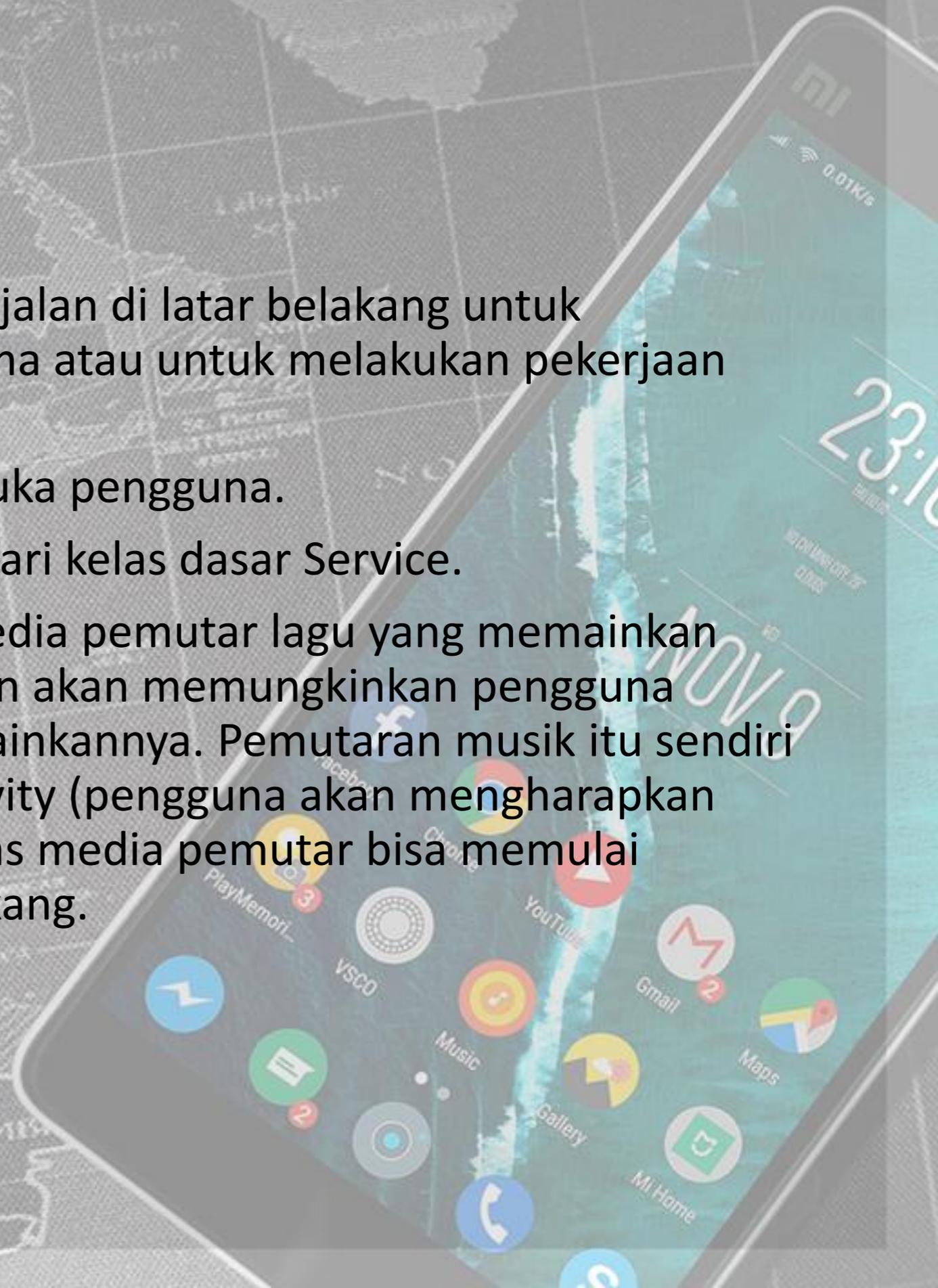
Activity...

- Android memiliki sejumlah tampilan siap pakai yang dapat digunakan - termasuk buttons, text fields, scroll bars, menu items, check boxes, dan banyak lagi.
- Hierarki tampilan ditempatkan dalam jendela aktivitas dengan metode `Activity setContentView ()`.
- Tampilan konten adalah objek Tampilan di akar hierarki.



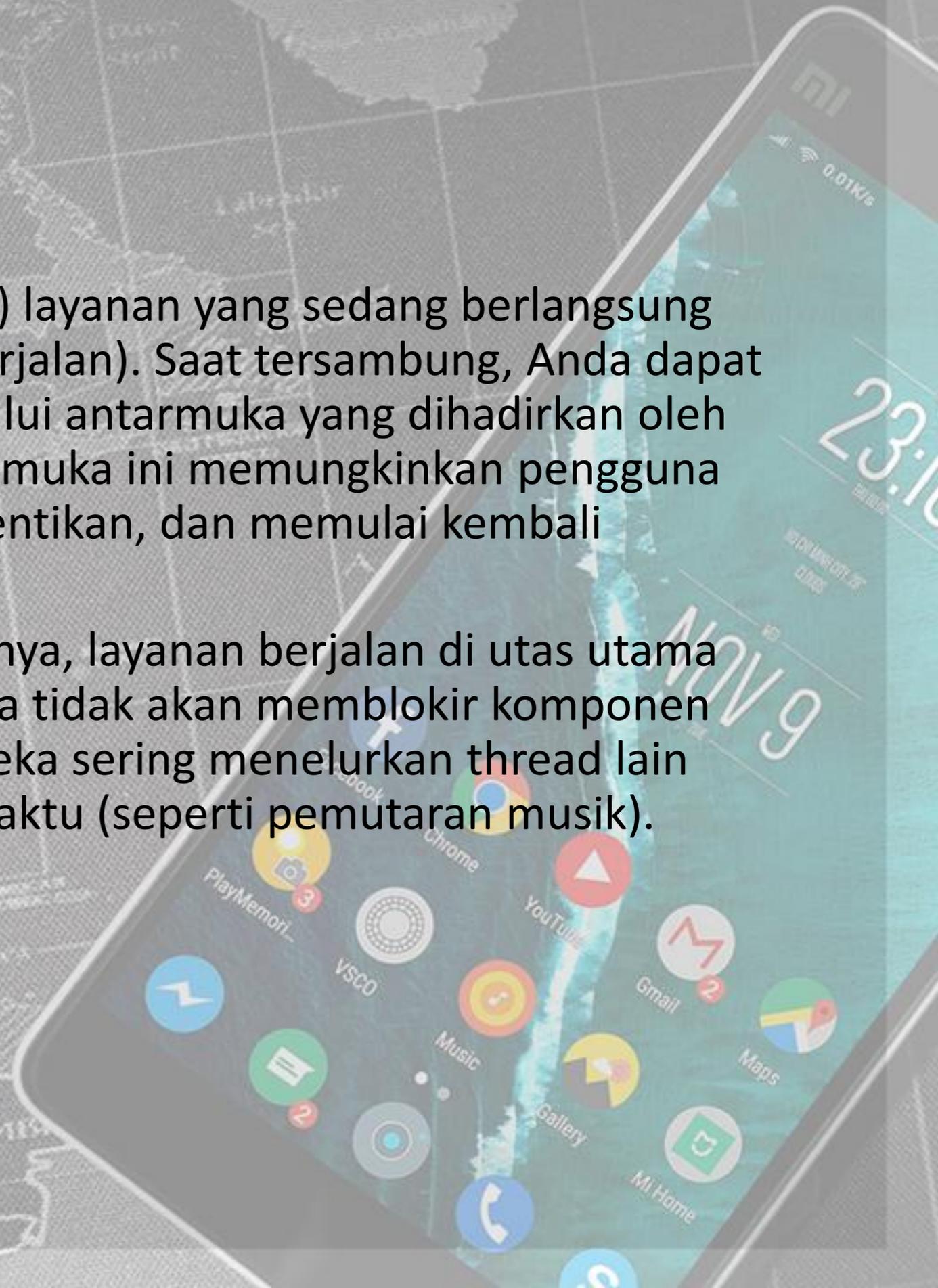
Service

- *Layanan* adalah komponen yang berjalan di latar belakang untuk melakukan operasi yang berjalan lama atau untuk melakukan pekerjaan bagi proses lama.
- Layanan tidak menyediakan antarmuka pengguna.
- Setiap service merupakan turunan dari kelas dasar Service.
- Contoh layanan yang baik adalah media pemutar lagu yang memainkan lagu. Satu atau lebih aktivitas pemain akan memungkinkan pengguna untuk memilih lagu dan mulai memainkannya. Pemutaran musik itu sendiri tidak akan ditangani oleh suatu activity (pengguna akan mengharapkan musik untuk terus bermain). Aktivitas media pemutar bisa memulai layanan untuk berjalan di latar belakang.



Service....

- Anda dapat terhubung ke (mengikat) layanan yang sedang berlangsung (dan memulai layanan jika belum berjalan). Saat tersambung, Anda dapat berkomunikasi dengan layanan melalui antarmuka yang dihadirkan oleh layanan. Untuk layanan musik, antarmuka ini memungkinkan pengguna untuk jeda, memutar ulang, menghentikan, dan memulai kembali pemutaran.
- Seperti aktivitas dan komponen lainnya, layanan berjalan di atas utas utama dari proses aplikasi. Sehingga mereka tidak akan memblokir komponen lain atau antarmuka pengguna, mereka sering menelurkan thread lain untuk tugas-tugas yang memakan waktu (seperti pemutaran musik).



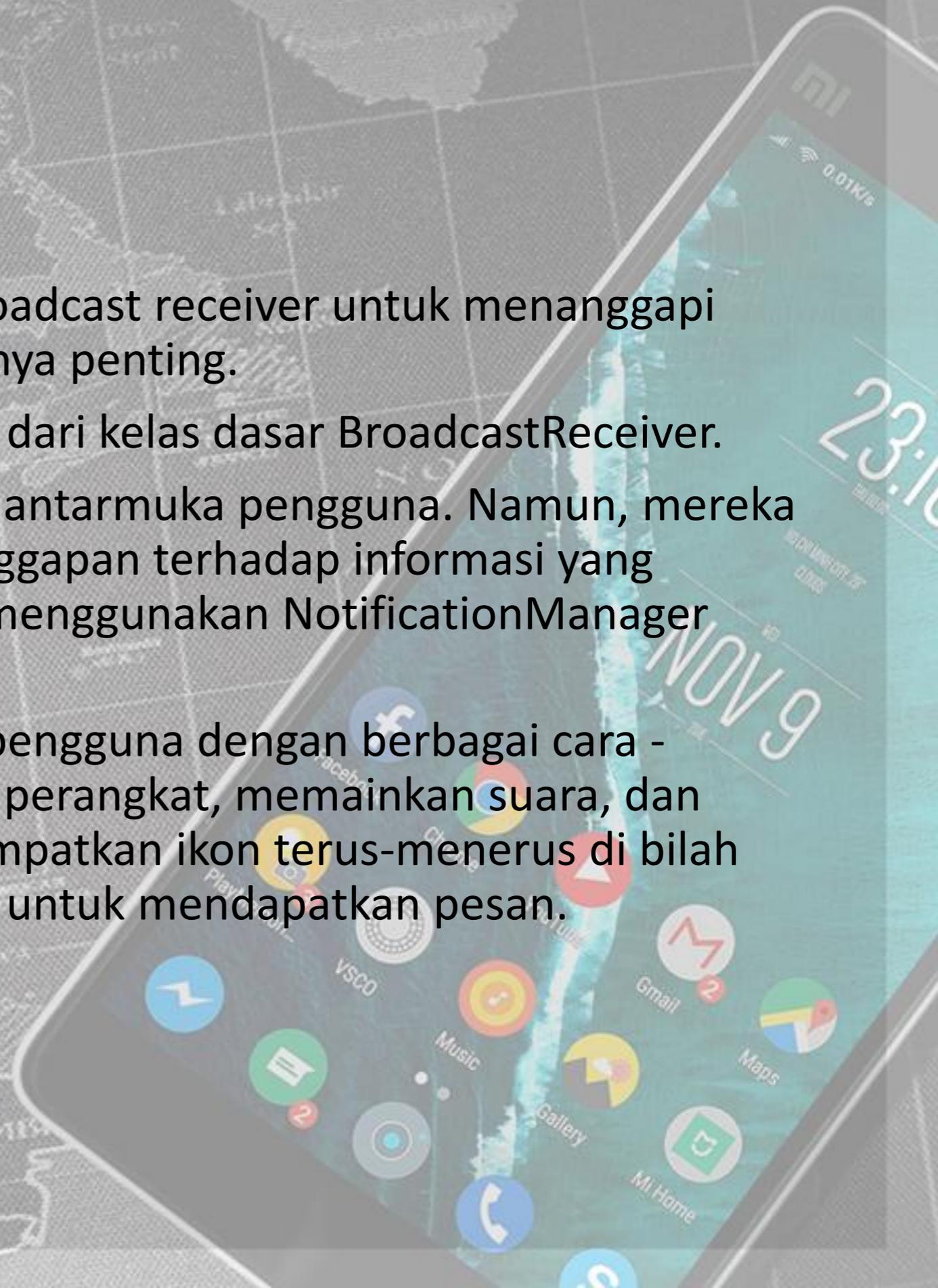
Broadcast Listener

- *Broadcast Listener* adalah komponen yang merespons dan bereaksi thd broadcast dalam lingkup sistem.
- Banyak broadcast yang berasal dari sistem—misalnya, siaran yang mengumumkan bahwa layar telah dinonaktifkan, baterai lemah, atau gambar telah direkam.
- Aplikasi juga bisa memulai siaran—misalnya untuk menginformasikan ke aplikasi lain bahwa sebagian data telah diunduh ke perangkat dan bisa digunakan aplikasi lain tersebut
- Penerima siaran diimplementasikan sebagai subkelas [BroadcastReceiver](#) dan setiap siaran dikirim sebagai objek.



Broadcast....

- Aplikasi dapat memiliki sejumlah broadcast receiver untuk menanggapi setiap pengumuman yang dianggapnya penting.
- Semua receiver merupakan turunan dari kelas dasar BroadcastReceiver.
- Penerima siaran tidak menampilkan antarmuka pengguna. Namun, mereka dapat memulai aktivitas sebagai tanggapan terhadap informasi yang mereka terima, atau mereka dapat menggunakan NotificationManager untuk mengingatkan pengguna.
- Notifikasi dapat menarik perhatian pengguna dengan berbagai cara - mem-flash backlight, menggetarkan perangkat, memainkan suara, dan sebagainya. Mereka biasanya menempatkan ikon terus-menerus di bilah status, yang dapat dibuka pengguna untuk mendapatkan pesan.



Content Provider

- *Content Provider* mengelola seperangkat data aplikasi bersama. Anda bisa menyimpan data dalam sistem file, database SQLite, di web, atau lokasi storage permanen lainnya yang bisa diakses aplikasi.
- Melalui ContentProvider materi, aplikasi lain bisa melakukan kueri atau bahkan memodifikasi data (jika penyedia materi mengizinkannya)
- ContentProvider merupakan turunan dari kelas dasar ContentProvider untuk menerapkan seperangkat metode standar yang memungkinkan aplikasi lain untuk mengambil dan menyimpan data dari jenis yang dikontrolnya.
- Aplikasi tidak memanggil metode ini secara langsung. Sebaliknya, mereka menggunakan objek ContentResolver dan memanggil metode sebagai gantinya.
- ContentResolver dapat berbicara dengan penyedia konten apa pun; itu bekerja sama dengan penyedia untuk mengelola komunikasi interproses yang terlibat.

Pustaka

- <https://developer.android.com/guide/?hl=id>
- android-developer-fundamentals-course-concepts-idn
- android-developer-fundamentals-course-practicals-idn

