



Mobile Programming

Catatan Kuliah #9

Alauddin Maulana Hirzan, M. Kom

0607069401

The background features a diagonal split between a teal upper-left section and a light gray lower-right section, with a white central area where the text is located.

Event, Event Listener dan Event Handling

The background consists of two large, overlapping geometric shapes. A teal-colored shape is in the upper-left corner, and a light gray shape is in the lower-left corner. The rest of the background is white. The word "Event" is centered in the white area.

Event



Event

Apa itu **Event** #1

Di Android, peristiwa mengacu pada sinyal atau notifikasi yang dihasilkan oleh sistem atau interaksi pengguna dengan perangkat. Peristiwa digunakan untuk memicu tindakan atau respons dalam aplikasi atau sistem operasi. **Event** juga bisa dikatakan sebagai hasil dari-pada interaksi pengguna dengan perangkat.

Terdapat berbagai macam jenis **Event** dalam Android:

1. Touch events
2. Key events
3. System events
4. App events



Event

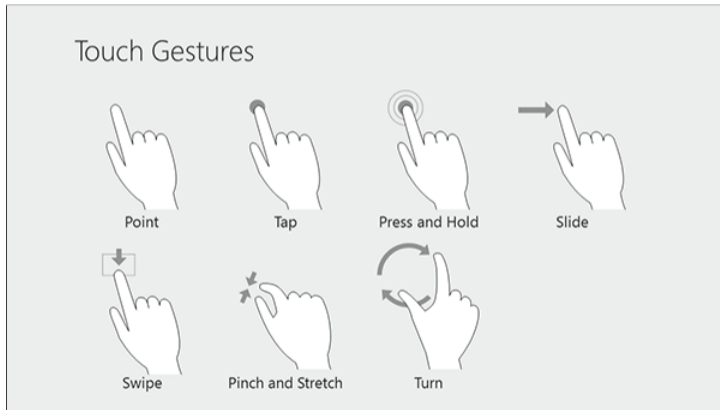
Apa itu **Event #2 - Touch Events**

Touch Events merupakan **Events** yang dihasilkan oleh sentuhan jari maupun **gesture**. Peristiwa ini dihasilkan ketika pengguna menyentuh layar, seperti mengetuk, mengusap, atau menggulir.

- ▶ Point
- ▶ Tap
- ▶ Press and Hold
- ▶ Slide
- ▶ Swipe
- ▶ Pinch and Stretch
- ▶ Turn

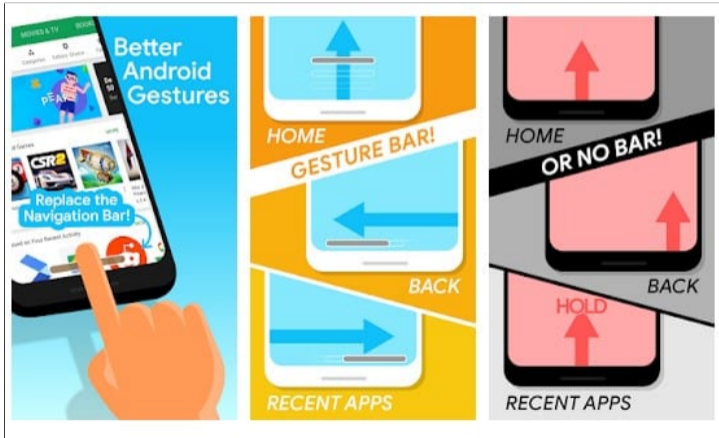
Event

Tampilan Touch Events



Event

Tampilan Touch Events





Event

Apa itu **Event #3 - Key Events**

Peristiwa ini dihasilkan ketika pengguna menekan atau melepaskan tombol fisik pada perangkat, seperti tombol volume atau tombol home.

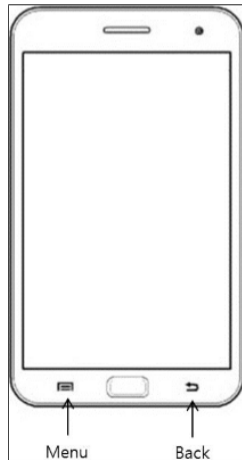
Jenis ini sangat bergantung dari jenis perangkat. Karena tidak semua dilengkapi dengan tombol yang sama. Beberapa perangkat memiliki tombol dedikasi Kamera, ada yang untuk Android Assistant dan lain-lain.

Event

Tampilan Key Events

Perangkat jaman dahulu memiliki tiga tombol utama di bagian depan dan 2 tombol samping untuk volume.

Namun berkat perkembangan jaman, tombol-tombol ini dihilangkan dan menyisakan 2 tombol volume dan 1 tombol power.



Event

Tampilan Key Events





Event

Apa itu **Event #4 - System Events**

Peristiwa ini dihasilkan oleh sistem operasi, seperti saat perangkat melakukan booting, saat level baterai berubah, atau saat panggilan telepon diterima. Sebagai contoh:

- ▶ Boot Selesai
- ▶ Baterai Diganti
- ▶ Layar Hidup/Mati
- ▶ Panggilan Masuk
- ▶ Panggilan Keluar
- ▶ SMS Diterima
- ▶ Wi-Fi Tersambung/Terputus
- ▶ Bluetooth Tersambung/Terputus
- ▶ Paket Dipasang/Dihapus
- ▶ Lokasi Berubah

Event

Apa itu **Event #5 - App Events**

Peristiwa ini dihasilkan oleh aplikasi itu sendiri, seperti ketika sebuah tombol diklik atau notifikasi diterima. Sehingga sifatnya hanya terbatas di konteks aplikasi saja.

- ▶ Aplikasi diluncurkan
- ▶ Aktivitas dimulai
- ▶ Tombol diklik
- ▶ Formulir terkirim
- ▶ Terjadi kesalahan
- ▶ Pemberitahuan diterima
- ▶ Pembelian dalam aplikasi dilakukan
- ▶ Aplikasi diperbarui
- ▶ Sesi dimulai/diakhiri
- ▶ Pengguna masuk/keluar

The background consists of two overlapping geometric shapes: a teal triangle in the top-left corner and a light gray triangle in the bottom-left corner, both pointing towards the center. The rest of the background is white.

Event Listener



Event Listener

Apa itu **EventListener** #1

Ketika **Event** itu terjadi, maka agar aplikasi dapat mendengarkan apabila terjadi atau tidak dapat dicapai dengan menggunakan **Listener**. Tanpa Listener, aplikasi tidak bisa mendengar apakah ada Event yang terjadi tau tidak. Untuk menempelkan sebuah objek dengan sebuah **Listener**, dapat dilakukan dengan mudah yaitu menambahkan **callbacks** seperti berikut:

- ▶ **OnClickListener**
- ▶ **OnLongClickListener**
- ▶ **TouchListener**
- ▶ **OnCheckedChangeListener**
- ▶ **TextWatcher**
- ▶ **SeekBarChangeListener**
- ▶ **OnItemSelectedListener**
- ▶ **OnScrollListener**



Event Listener

Apa itu **EventListener** #2 - Listener Click

- ▶ **OnClickListener**: Pendengar ini digunakan untuk mendeteksi ketika pengguna mengklik sebuah tampilan, seperti tombol atau gambar. Ini diimplementasikan dengan menggunakan metode `setOnClickListener()`.
- ▶ **OnLongClickListener**: Pendengar ini digunakan untuk mendeteksi ketika pengguna menekan lama sebuah tampilan. Diimplementasikan dengan menggunakan metode `setOnLongClickListener()`.
- ▶ **OnMenuItemClickListener**: Pendengar ini digunakan untuk mendeteksi ketika pengguna mengklik sebuah item menu di menu pilihan atau menu konteks. Diimplementasikan dengan menggunakan metode `setOnMenuItemClickListener()`.



Event Listener

Apa itu **EventListener** #3 - Listener Click

- ▶ **OnNavigationItemSelectedListener:** Pendengar ini digunakan untuk mendeteksi ketika pengguna mengklik sebuah item dalam laci navigasi atau tampilan navigasi bawah. Ini diimplementasikan menggunakan metode `setNavigationItemSelectedListener()` atau metode `setOnNavigationItemSelectedListener()`.
- ▶ **OnTabClickListener:** Pendengar ini digunakan untuk mendeteksi ketika pengguna mengklik sebuah tab di `TabLayout`. Diimplementasikan dengan menggunakan metode `addOnTabSelectedListener()`.



Event Listener

Apa itu **EventListener** #4 - Listener Database

- ▶ **ValueEventListener**: Pendengar ini digunakan dengan Firebase Realtime Database
- ▶ **OnDataChangeListener**: Pendengar ini digunakan dengan pustaka Room persistence
- ▶ **ContentObserver**: Pendengar ini digunakan dengan Penyedia Konten Android
- ▶ **CursorAdapter**: Pendengar ini digunakan dengan basis data SQLite



Event Listener

Apa itu **EventListener** #5 - Listener Gesture

- ▶ **GestureDetector.OnGestureListener:** Pendengar ini digunakan untuk mendeteksi gerakan seperti usapan, ketukan, dan ketukan dua kali. Ini diimplementasikan dengan menggunakan metode `onTouchEvent()`.
- ▶ **ScaleGestureDetector.OnScaleGestureListener:** Pendengar ini digunakan untuk mendeteksi gerakan penskalaan seperti pinch-to-zoom. Ini diimplementasikan menggunakan metode `onTouchEvent()`.
- ▶ **GestureDetector.OnDoubleTapListener:** Pendengar ini digunakan untuk mendeteksi gerakan ketuk dua kali. Diimplementasikan dengan menggunakan metode `onDoubleTap()`.



Event Listener

Apa itu **EventListener** #6 - Listener Gesture

- ▶ **View.OnTouchListener:** Pendengar ini digunakan untuk mendeteksi berbagai gerakan sentuh, termasuk seret, lempar, dan tekan lama. Ini diimplementasikan menggunakan metode `onTouch()`.
- ▶ **OnGesturePerformed:** Pendengar ini digunakan untuk mendeteksi gerakan yang ditentukan pengguna. Diimplementasikan dengan menggunakan metode `onGesturePerformed()`.

The background consists of two large, overlapping geometric shapes. A teal-colored shape is in the upper-left corner, and a light gray shape is in the lower-left corner. The rest of the background is white. The text 'Event Handler' is centered in the white area.

Event Handler



Event Handler

Apa itu **EventHandler** #1

Sebuah fungsi yang memiliki tugas untuk menindaklanjuti Event yang terjadi. Jika tombol ditekan atau Event lainnya, maka kode ini akan dipanggil sebagai bentuk respon.

Terdapat tiga cara dalam mengimplementasikan kode ini jika menggunakan Bahasa Java maupun Kotlin.



Event Handler

Apa itu **EventHandler** #2 - Implementasi Handling Java

Kode

```
Button myButton = findViewById(R.id.my_button);

myButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Code to handle the click event
    }
});
```



Event Handler

Apa itu **EventHandler** #3 - Implementasi Handling Kotlin

Kode

```
val myButton = findViewById<Button>(R.id.my_button)

myButton.setOnClickListener {
    // Code to handle the click event
}
```

Kode Kotlin terlihat lebih pendek namun memiliki fungsi yang sama.



Event Handler

Apa itu **EventHandler** #3 - Implementasi Handling Kotlin 2

Kode

```
override fun onCreate(savedInstanceState: Bundle?) {  
    button1 = findViewById(R.id.button1)  
    button1.setOnClickListener(this)  
}  
override fun onClick(view: View) {  
    when (view.id) {  
        R.id.button1 -> {  
            // Handle button 1 click event  
        }  
    }  
}}
```




Event Handler

Apa itu **EventHandler** #4 - Implementasi Gesture Handling

Kode

```
class CustomGestureListener : GestureDetector.SimpleOnGestureListener() {  
    override fun onDown(event: MotionEvent): Boolean {  
        // Handle the "down" event  
        return true  
    }  
  
    override fun onFling(event1: MotionEvent, event2: MotionEvent, velocityX: Float,  
velocityY: Float): Boolean {  
        // Handle the "fling" event  
        return true  
    }  
}
```

