



BINAR
ACADEMY

Dialog di Android

Gold - Chapter 4 - Topic 2

**Selamat datang di Chapter 4 Topic 2 *online course* Android
Developer dari Binar Academy!**





Hai teman-teman ✨

Wuih balik lagi nih kita ketemu di topik baru. Pada topik sebelumnya, kita udah belajar tentang Toast dan Snackbar.

Di topik kali ini, kita eksplor tentang tata cara pembuatan **Dialog di Android**, yang berperan dalam membangun interaksi antara aplikasi dan pengguna-nya.

Kayak apa sih? Yuk langsung aja kita OTW!



Detailnya, kita bakal bahas hal-hal berikut ini:

- AlertDialog
- Cara Membuat Alert Dialog
- Kapan menggunakan Alert Dialog

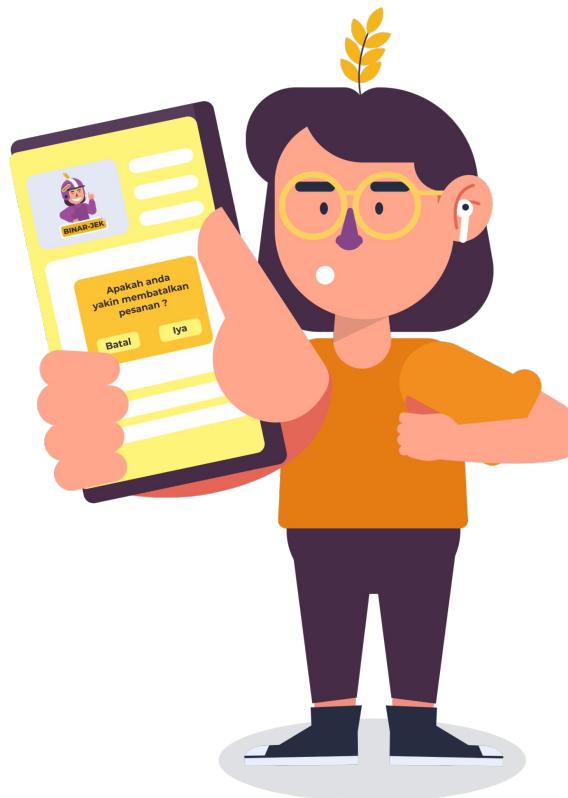




Dialog? Baru denger banget.

Kalo diterjemahin harusnya sih dialog yang berisi info penting ya.. tapi apa bener kayak gitu artinya?

Apa sih yang dimaksud dengan **Dialog**?

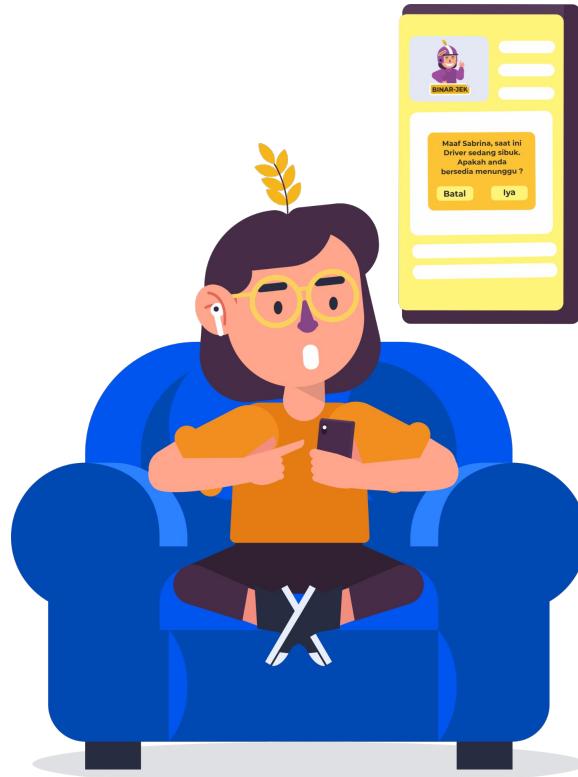


Apa itu Dialog?

Yepp, AlertDialog itu sama aja dengan Dialog. Cuma kalo di Android kita nyebutnya **AlertDialog**.

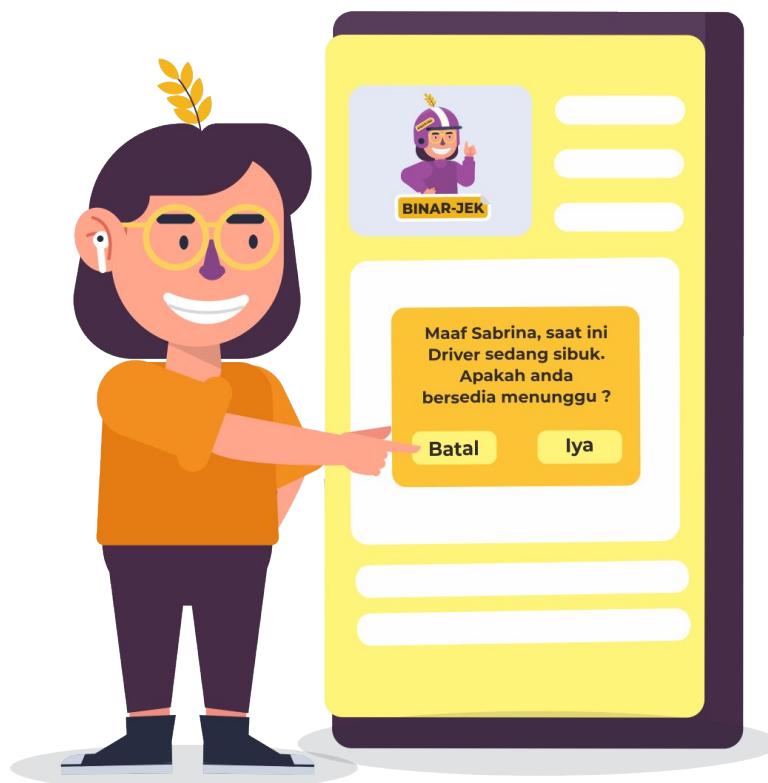
AlertDialog merupakan sebuah **jendela pop-up** untuk menampilkan pesan informasi.

Selain itu, AlertDialog juga bisa ditambah dengan adanya beberapa tombol aksi.



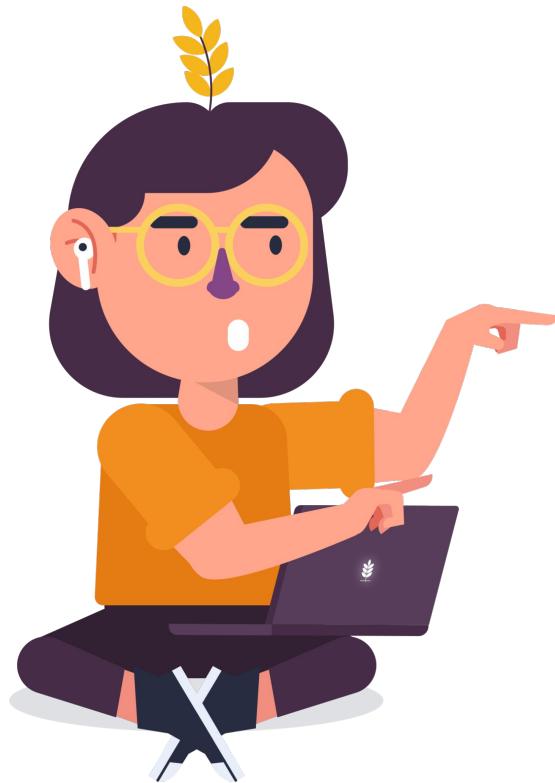
AlertDialog biasanya digunakan untuk menampilkan pesan informasi yang cukup penting, yang mengharuskan user untuk membaca pesan tersebut.

Misalnya tuh pesan untuk konfirmasi, pemberitahuan error atau kesalahan, dan pesan lainnya.



Kalo sebelumnya kita sebut Toast dan Snackbar sebagai pembawa informasi juga, AlertDialog ini memiliki perbedaan dengan Toast maupun Snackbar itu, yaitu :

“AlertDialog tidak bisa hilang atau menutup secara otomatis.”



Jenis-jenis AlertDialog

Selain nggak bisa ilang secara otomatis, AlertDialog juga punya beberapa jenis. Kayak gini :

- **AlertDialog** **Standard**
AlertDialog tanpa adanya tombol aksi
- **AlertDialog Standard dengan Tombol Aksi** **Standard with Action Buttons**
AlertDialog dengan tombol aksi



- **AlertDialog dengan Custom Layout**
AlertDialog dengan Layout yang bisa kita tentukan
- **AlertDialog dengan DialogFragment**
AlertDialog dengan menggunakan class DialogFragment.

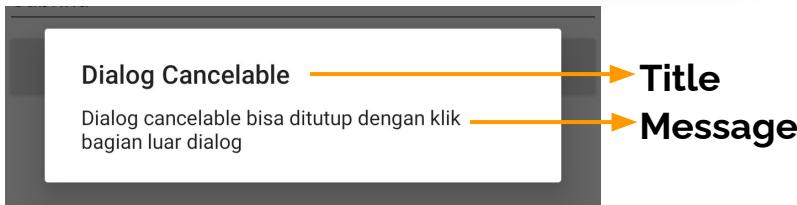


Ciee, udah tau definisi AlertDialog dan jenis-jenisnya~

**Selanjutnya kita coba praktik yuk!
Cuss kita bahas cara membuat buat Dialog~**



```
val dialog = AlertDialog.Builder(this)
dialog.setTitle("Dialog Cancelable")
dialog.setMessage("Dialog cancelable bisa
ditutup dengan klik bagian luar dialog")
dialog.setCancelable(true)
dialog.show()
```



AlertDialog Standard

Fun fact, AlertDialog ini yang paling simpel dan gampang banget buatnya. Iya, coba kamu cek kode disamping!

AlertDialog ini nggak punya tombol aksi apapun, termasuk tombol untuk menutupnya.

Itu kenapa kita set AlertDialog ini **cancelable**-nya menjadi **true**, sehingga dialog ini akan tertutup jika user melakukan klik di bagian luar dialog.



```
val dialog = AlertDialog.Builder(this)
dialog.setTitle("Dialog Cancelable")
dialog.setMessage("Dialog cancelable bisa
ditutup dengan klik bagian luar dialog")
dialog.setCancelable(true)
dialog.show()
```

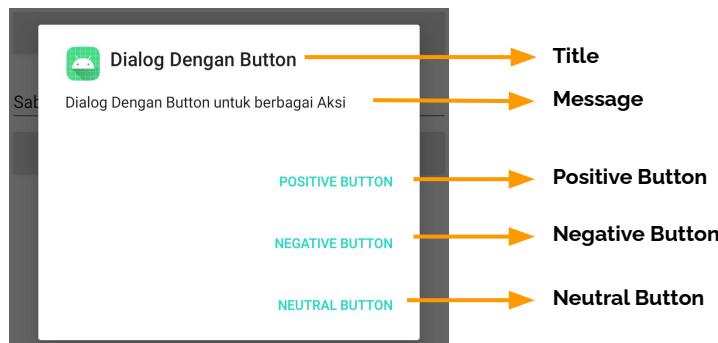


Kenapa Cancelable-nya di set jadi True?

Iya karena kalo Cancelable nya dijadiin false, nanti jadinya
nggak bisa ketutup gitu dialognya~



```
● ● ●  
val dialog = AlertDialog.Builder(this)  
dialog.setTitle("Dialog Dengan Button")  
dialog.setMessage("Dialog Dengan Button untuk berbagai Aksi")  
dialog.setIcon(R.mipmap.ic_launcher)  
  
dialog.setCancelable(false)  
dialog.setPositiveButton("Positive Button"){dialogInterface, p1 ->  
    Toast.makeText(this,"Positive Button Clicked",Toast.LENGTH_LONG).show()  
}  
  
dialog.setNegativeButton("Negative Button"){dialogInterface, p1->  
    Toast.makeText(this,"Negative Button Clicked",Toast.LENGTH_LONG).show()  
}  
  
dialog.setNeutralButton("Neutral Button"){dialogInterface, p1->  
    Toast.makeText(this,"Neutral Button Clicked",Toast.LENGTH_LONG).show()  
}  
  
dialog.show()
```



AlertDialog dengan Tombol Aksi

AlertDialog juga bisa menyediakan berbagai tombol aksi. Tombol aksi pada AlertDialog ada tiga jenis, yaitu :

- **Positive Button.** Tombol untuk tindakan yang sifatnya positif atau menyetujui dengan pesan yang ada pada AlertDialog.

Misalnya seperti "OK", "Lanjut", "Ya".



- **Negative Button** : Tombol untuk tindakan yang bertentangan dengan sifatnya positif. Negative Button biasanya ada jika Positive Button juga ada.

Contohnya seperti “Deny”, “Tolak”, “Cancel”, “Tidak”.

- **Neutral Button** : Tombol untuk tindakan yang netral. Tidak positif, juga tidak negatif.

Neutral Button biasanya ada jika Negative dan Positive Button juga ada.



Hayoo.. Kalo kamu liat gambar disamping udah kebayang belum yang mana Positive, Negative dan Neutral Button?

Iya kamu bener! untuk pesan “Apakah Anda menyukai aplikasi ini?” memiliki Positive Button “Iya”, Negative Button “Tidak”, dan Neutral Button “Memilih untuk tidak menjawab” atau “Lain Kali”.

Untuk ketiga jenis tombol tersebut, kalo user memilih salah satu, maka AlertDialog akan ter **dismiss** atau akan menutup secara otomatis.

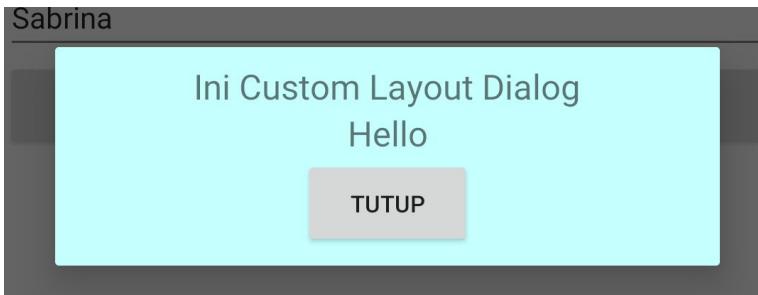


```
val view = LayoutInflater.from(this).inflate(R.layout.custom_dialog,null,false)
val dialogBuilder = AlertDialog.Builder(this)
dialogBuilder.setView(view)

val dialog = dialogBuilder.create()

view.btnTutup.setOnClickListener {
    Toast.makeText(this,"Custom Dialog Closed",Toast.LENGTH_LONG).show()
    dialog.dismiss()
}

dialog.show()
```



AlertDialog dengan Custom Layout

AlertDialog ini cocok buat yang pingin membuat AlertDialog dengan tampilan sesuai dengan yang diinginkan.

Ini memungkinkan karena AlertDialog bisa juga menggunakan file XML yang ada pada resource layout sebagai tampilan kontennya.

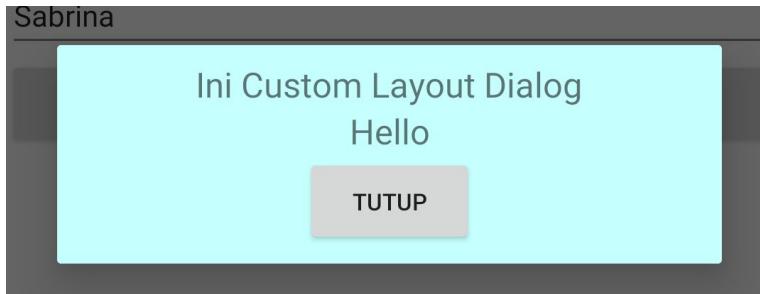


```
val view = LayoutInflater.from(this).inflate(R.layout.custom_dialog,null,false)
val dialogBuilder = AlertDialog.Builder(this)
dialogBuilder.setView(view)

val dialog = dialogBuilder.create()

view.btnExit.setOnClickListener {
    Toast.makeText(this,"Custom Dialog Closed",Toast.LENGTH_LONG).show()
    dialog.dismiss()
}

dialog.show()
```



Wah kalo gitu, tampilan dialognya bisa dikreasikan sesuka kita dong?

Betul banget! Kita bisa menambahkan Widget View dan menatanya dengan ViewGroup apapun! Kita juga bisa nambahin Tombol, TextView, EditText dan sebagainya~



```
● ● ●

class ExampleDialogFragment() : DialogFragment() {
    lateinit var name: String

    //Secondary Constructor untuk menangani jika DialogFragment membutuhkan data custom
    constructor(name: String):this(){
        this.name = name
    }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_dialog, container, false)
    }

    override fun onResume() {
        super.onResume()
        dialog?.window?.setLayout(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT)
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        //Cek apakah variable name sudah diinisialisasi?
        if(this::name.isInitialized){
            tvHello.text = name
        }

        btnTutup.setOnClickListener {
            Toast.makeText(activity,"DialogFragment Ditutup dari dalam fragment",Toast.LENGTH_LONG).show()
            dialog?.dismiss()
        }
    }
}
```

AlertDialog dengan DialogFragment

Nah, kalo ini, bikin dialognya pake DialogFragment!

Artinya, kita membuat class terpisah yang meng-extend class DialogFragment.

Mirip kayak bikin Fragment ya?



```
● ● ●

class ExampleDialogFragment() : DialogFragment() {
    lateinit var name: String

    //Secondary Constructor untuk menangani jika DialogFragment membutuhkan data custom
    constructor(name: String):this(){
        this.name = name
    }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_dialog, container, false)
    }

    override fun onResume() {
        super.onResume()
        dialog?.window?.setLayout(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT)
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        //Cek apakah variable name sudah diinisialisasi?
        if(this::name.isInitialized){
            tvHello.text = name
        }

        btnTutup.setOnClickListener {
            Toast.makeText(activity,"DialogFragment Ditutup dari dalam fragment",Toast.LENGTH_LONG).show()
            dialog?.dismiss()
        }
    }
}
```

Nggak cuma cara bikinnya yang mirip, sifat DialogFragment juga mirip sama Fragment. Bedanya, DialogFragment ditampilkan dalam bentuk Dialog yang lebih ke pop-up.

Sedangkan Fragment umumnya lebih ke menempel, menjadi bagian dari suatu Activity.



AlertDialog dengan DialogFragment

Balik lagi ke laptop, AlertDialog ini kan buatnya pake DialogFragment. Dimana kita perlu membuat class terpisah yang meng-extend class DialogFragment.

Nah setelah itu udah jadi, kita bisa memanggilnya dengan method show, dari object DialogFragment yang udah kita buat.

Jadi deh. Gampang kan?

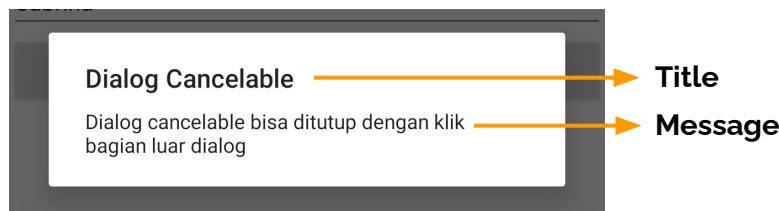


Itu dia cara buat AlertDialog~

Tadi siapa yang masih inget kalo AlertDialog punya banyak jenis?

Dari jenis itu gimana nih kita bisa **Nentuin dialog** yang tepat pada kondisi tertentu?

Jawabanya ada di slide selanjutnya~

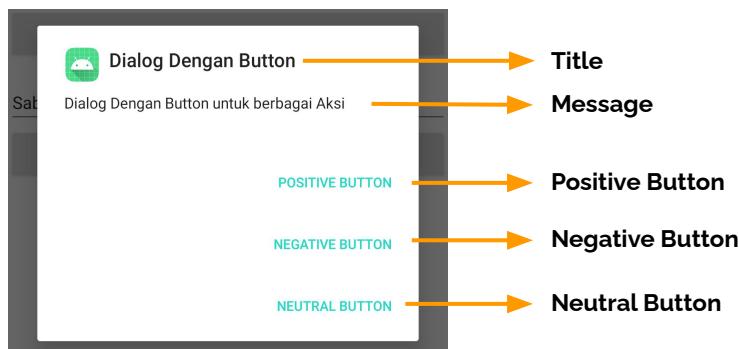


Kapan menggunakan AlertDialog Standard?

Biasanya, AlertDialog ini digunakan untuk pesan berisi hint atau pesan yang tidak membutuhkan aksi.

Sesuai dengan namanya, jenis ini punya beberapa karakter nih :

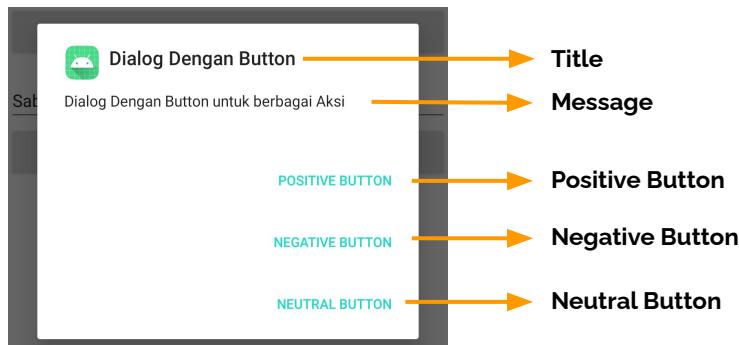
- Pesan yang ditampilkan cukup hanya dengan adanya title, message, dan tambahan icon.
- Tidak perlu adanya tombol Action



Kapan menggunakan AlertDialog dengan Tombol Aksi?

Ketika pesan informasi membutuhkan setidaknya satu aksi. Kombinasi adanya tombol aksi yaitu :

- Positive Button
- Positive Button + Negative Button
- Positive Button + Negative Button + Neutral Button

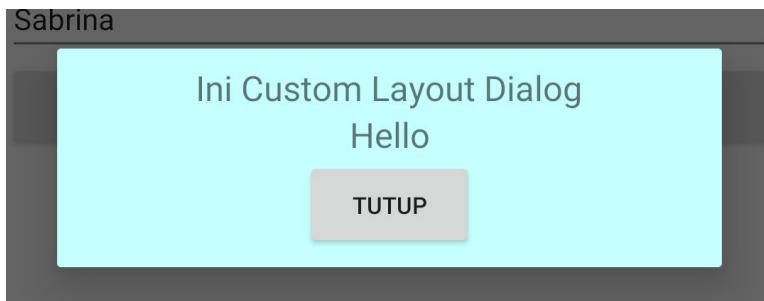


Nggak cuma itu, kita juga bisa pake AlertDialog dengan tombol Aksi ketika pesan informasi yang disampaikan membutuhkan aksi user.

Untuk itulah biasanya AlertDialog dengan tombol aksi ini memiliki atribut ***cancelable = false***.

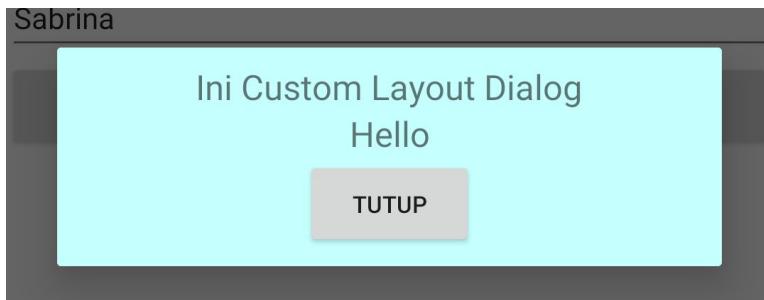


Sabrina



Kapan menggunakan AlertDialog dengan Custom Layout?

Seperti yang kita tahu, AlertDialog ini cocok buat kamu yang pengen bikin AlertDialog dengan tampilan yang sesuai dengan keinginanmu.



Jadi, kita bisa menggunakan AlertDialog ini jika:

- Menginginkan tampilan custom
- Tidak terlalu banyak layout ataupun perintah yang bisa dijalankan dari dalam dialog.



Gimana Penggunaan AlertDialog dengan DialogFragment?

Seperti yang kita ketahui kalau DialogFragment itu mirip sama Fragment, sehingga kita bisa menggunakan DialogFragment jika :

- **Terdapat banyak Layout dan perintah yang dapat dijalankan dari dalam Dialog.**

Karena code perintah terdapat pada class yang meng extend DialogFragment. Jadi Activity atau Fragment kita bisa lebih sedikit code nya.



- **Jika membutuhkan Fragment Lifecycle.**

Karena DialogFragment adalah turunan dari Fragment sehingga Fragment Lifecycle juga berlaku di DialogFragment



Mau nyobain contoh Project Aplikasi untuk Dialog ini?

Boleeeh~ Silakan untuk clone project dari lik di bawah ini yes~~

<https://github.com/sennohananto/Chapter5Dialog>



Biar makin asik, yuk kita ikut Challenge ini!

Buatlah suatu Aplikasi dengan fitur:

- Menggunakan AlertDialog Standard
- Menggunakan AlertDialog dengan tombol Aksi
- Menggunakan AlertDialog Custom Layout
- Menggunakan DialogFragment

Pengumpulan tugas diserahkan kepada mentor.

Kita akan membahas bersama-sama di pertemuan berikutnya.

Saatnya kita Quiz!





1. Class apa yang bisa digunakan untuk membuat Dialog?

- A. Activity
- B. AlertDialog
- C. WarningDialog



1. Class apa yang bisa digunakan untuk membuat Dialog?

- A. Activity
- B. AlertDialog
- C. WarningDialog

Kita bisa membuat tampilan dialog dengan kelas bernama AlertDialog yes!



2. Apa nama atribut yang memungkinkan suatu AlertDialog bisa ditutup dengan klik bagian luar AlertDialog?

- A. Cancelable
- B. Dismissible
- C. Closeable

2. Apa nama atribut yang memungkinkan suatu AlertDialog bisa ditutup dengan klik bagian luar AlertDialog?

- A. Cancelable
- B. Dismissible
- C. Closeable

Atribut cancelable bisa diisi dengan true atau false.

Jika true, dialog bisa ditutup dengan cara user klik di bagian luar dialog. Sedangkan jika false, dialog tidak akan tertutup walaupun user klik di bagian luar dialog.



3. Dalam membuat AlertDialog dengan Layout Custom, kita membutuhkan class LayoutInflater. Apa fungsi LayoutInflater?

- A. Mendapatkan View dari Layout XML
- B. Mendapatkan Dialog
- C. Menampilkan Dialog

3. Dalam membuat AlertDialog dengan Layout Custom, kita membutuhkan class LayoutInflater. Apa fungsi LayoutInflater?

- A. Mendapatkan View dari Layout XML
- B. Mendapatkan Dialog
- C. Menampilkan Dialog

LayoutInflater berfungsi sebagai “Peniup” layout dari resource XML. Hasil ‘Tiupan’ LayoutInflater berupa object View.

Object view inilah yang bisa kita pakai sebagai Layout dari AlertDialog.



4. Perhatikan sintaks berikut!

Kode yang disusun bertujuan untuk

- A. Membuat AlertDialog dengan Tombol Aksi
- B. Membuat Dialog Fragment
- C. Membuat Custom Layout pada AlertDialog

```
val dialog = AlertDialog.Builder(this)
dialog.setTitle("Dialog Dengan Button")
dialog.setMessage("Dialog Dengan Button untuk berbagai Aksi")
dialog.setIcon(R.mipmap.ic_launcher)

dialog.setCancelable(false)
dialog.setPositiveButton("Positive Button"){dialogInterface, p1 ->
    Toast.makeText(this,"Positive Button Clicked",Toast.LENGTH_LONG).show()
}

dialog.setNegativeButton("Negative Button"){dialogInterface, p1->
    Toast.makeText(this,"Negative Button Clicked",Toast.LENGTH_LONG).show()
}

dialog.setNeutralButton("Neutral Button"){dialogInterface, p1->
    Toast.makeText(this,"Neutral Button Clicked",Toast.LENGTH_LONG).show()
}

dialog.show()
```

4. Perhatikan sintaks berikut!

Kode yang disusun bertujuan untuk

- A. Membuat AlertDialog dengan Tombol Aksi
- B. Membuat Dialog Fragment
- C. Membuat Custom Layout pada AlertDialog

```
val dialog = AlertDialog.Builder(this)
dialog.setTitle("Dialog Dengan Button")
dialog.setMessage("Dialog Dengan Button untuk berbagai Aksi")
dialog.setIcon(R.mipmap.ic_launcher)

dialog.setCancelable(false)
dialog.setPositiveButton("Positive Button"){dialogInterface, p1 ->
    Toast.makeText(this,"Positive Button Clicked",Toast.LENGTH_LONG).show()
}

dialog.setNegativeButton("Negative Button"){dialogInterface, p1->
    Toast.makeText(this,"Negative Button Clicked",Toast.LENGTH_LONG).show()
}

dialog.setNeutralButton("Neutral Button"){dialogInterface, p1->
    Toast.makeText(this,"Neutral Button Clicked",Toast.LENGTH_LONG).show()
}

dialog.show()
```

Sintaks ini akan Tombol aksi yang ada pada AlertDialog yaitu Positive Button, Negative Button, Neutral Button.



5. Class mana yang merupakan turunan dari class Fragment dan memungkinkan suatu AlertDialog bertindak layaknya Fragment?

- A. FragmentDialog
- B. DialogFragment
- C. DialogAlert

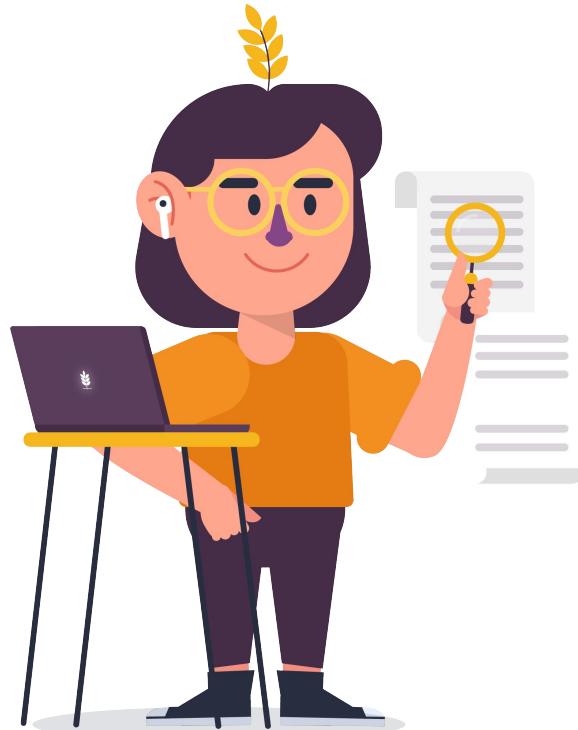
5. Class mana yang merupakan turunan dari class Fragment dan memungkinkan suatu AlertDialog bertindak layaknya Fragment?

- A. FragmentDialog
- B. DialogFragment
- C. DialogAlert

Jika suatu class meng extend class DialogFragment, class tersebut bisa dijadikan Dialog yang memiliki kriteria Fragment!

Referensi dan bacaan lebih lanjut~

1. [AlertDialog | Android Developers](#)
2. [DialogFragment | Android Developers](#)





Nah, selesai sudah pembahasan kita di Chapter 4 Topic 2 ini.

Selanjutnya, kita bakal belajar cara membuat tampilan Recycler View.

Penasaran kayak gimana? Cus langsung ke topik selanjutnya~



Terima Kasih!



Next Topic

loading...