



Shared Preferences

Gold - Chapter 4 - Topic 4

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





Hai teman-teman ✨

Di topik sebelumnya, kita sudah eksplor banyak tentang bentuk tampilan RecyclerView.

Mulai di topik kali ini, kita coba belajar metode penyimpanan data aplikasi, **SharedPreferences**.

Kayak apa sih doi? yuk, lanjut!





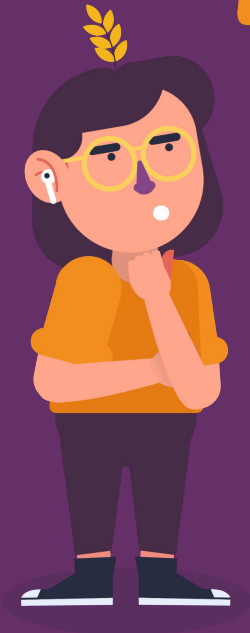
Detailnya, kita bakal bahas hal-hal berikut ini:

- Pengenalan Shared Preference
- Cara Deklarasi Shared Preference
- Praktik Penggunaan Shared Preference





SharedPreferences



Android menyediakan banyak cara untuk menyimpan data di suatu aplikasi.

Salah satu cara ini disebut dengan **SharedPreferences**.

First thing first, kita harus tau dulu dong apa yang dimaksud dengan **SharedPreferences**?



Apa itu SharedPreferences?

SharedPreferences adalah metode yang memungkinkan kita **menyimpan dan mengambil data dalam bentuk key dan value**.

Data yang disimpan dalam aplikasi bakal tetap ada meskipun aplikasi ditutup, dihapus, atau dibersihkan (lewat penyimpanan lokal).

SharedPreferences

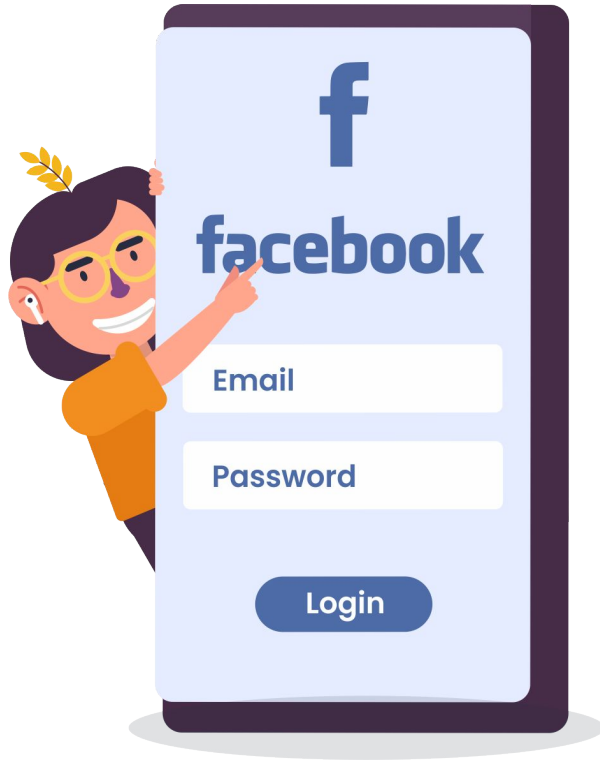




File pengaturan Android menggunakan sharedPreferences untuk menyimpan data pengaturan aplikasi dalam bentuk file XML, tepatnya ada di dalam:

data/data/{application package}/share_prefs direktori





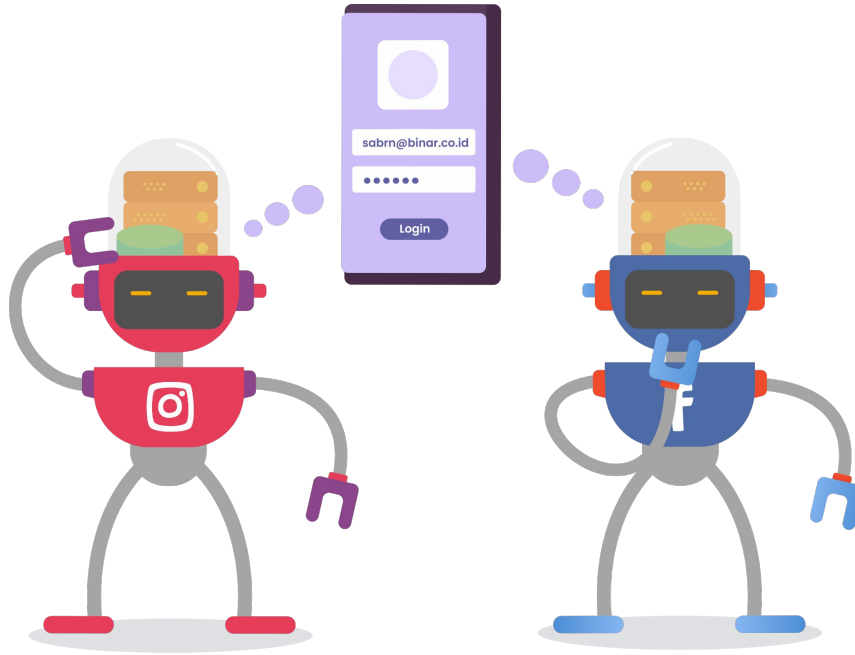
Usut punya usut, kita tuh udah sering banget menggunakan “SharedPreferences” di Aplikasi yang kita gunakan sehari-hari.

Coba deh, kamu pernah login di aplikasi Facebook atau Instagram belum?



Kalau kamu inget-inget, pas kamu tutup aplikasi, lalu membuka aplikasi lagi, **kamu nggak perlu untuk login lagi, kan?**

Padahal waktu pertama kamu install aplikasi, kamu harus login dulu. Tapi kalo sekedar tutup dan buka aplikasi, ya.. nggak perlu login lagi.



Nah, dibalik dari apa yang kamu alami itu ada proses kayak gini..

Aplikasi Facebook dan Instagram sedang “mengingat”, kalau kamu pernah *login* di aplikasi tersebut dengan menggunakan “SharedPreferences”.



Wuihhh, keren nggak?

Sekarang jadi pingin coba eksplor
SharedPreferences nggak sie, bestie?

Langsung aja yuk kita caw buka cara
menggunakan **SharedPreferences**~



Mengakses SharedPreferences

Untuk mengakses SharedPreferences ke dalam aplikasi kita, kita perlu mendapatkan instance untuk menggunakan salah satu method, yakni:

getSharedPreferences ()

Pada gambar samping merupakan salah satu contoh penerapan dari SharedPreferences.

```
val sharedPreferences: SharedPreferences =  
this.getSharedPreferences(String preferences_fileName,int mode)
```


preferences_fileNames adalah nama file dari SharedPreferences dan mode operasional file.



Mengedit SharedPreferences

Untuk mengubah-ubah SharedPreferences, kita bisa menggunakan object SharedPreferences.Editor.

Pada gambar disamping, itu merupakan contoh pembuatan editor SharedPreferences:



```
val editor:SharedPreferences.Editor = sharedPreferences.edit()
```



Nah kalo buat menghapus data preferensi aplikasi, kita bisa tuh pake dua cara:

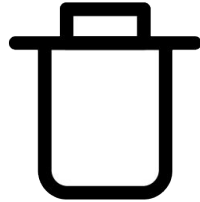
- Pertama, memanggil method `editor.remove("key")` yang berfungsi untuk menghapus key value yang ditentukan.
- Kedua, pake method `editor.clear()`, yang berfungsi untuk menghapus semua data preferensi.



```
val editor:SharedPreferences.Editor = sharedPreferences.edit()
```

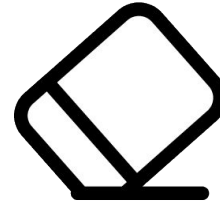


SharedPreferences akan otomatis terhapus apabila terjadi dua hal ini:



Uninstall

Meng-uninstall aplikasi.



Clear data

Menghapus data aplikasi melalui pengaturan.



Hmm.. biar makin kebayang cara pake SharedPreferences, kita **coba buat coding** dari beberapa contoh yang udah ada aja.

Gimana menurut kalian? siapa?



Contoh Aplikasi Shared Preference

Dalam contoh ini, kita bakal bisa menginput data (id dan name) dari EditText dan menyimpannya dalam SharedPreferences.

The screenshot shows a mobile application interface with a purple header bar. Below the header, there are two input fields: 'Masukkan Id' with the value '112' and 'Masukkan Nama' with the value 'Sabrina'. Below these fields, there are two labels: 'Idmu' with the value '112' and 'Namamu' with the value 'Sabrina'. At the bottom of the screen, there are three buttons: 'SAVE', 'VIEW', and 'CLEAR'.

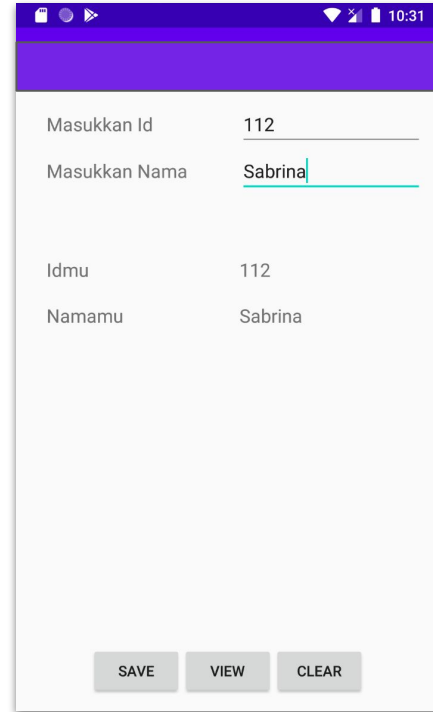
Masukkan Id	112
Masukkan Nama	Sabrina
Idmu	112
Namamu	Sabrina

SAVE VIEW CLEAR



SharedPreferences diambil dan ditampilkan dalam TextView dengan melakukan klik pada Tombol. Selain itu ada juga tombol untuk menghapus SharedPreferences.

Sebelum itu kita persiapkan dulu Android Studio-nya. Kita buat project dengan nama MySharedPreferences.

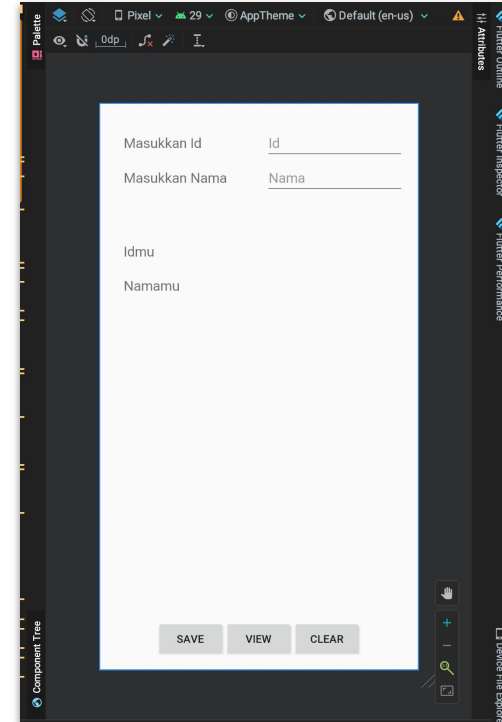




Desain Layout XML dari MainActivity bakal tampil kayak preview disamping. Kalo udah, kamu kasih ID setiap View, lalu disesuaikan aja.

Untuk mudahnya, kamu bisa menggunakannya dari code Layout XML yang udah kita sediakan di link berikut:

[XML Code Klink Disini](#)





Pada MainActivity.kt, kita dapatkan SharedPreferences dengan cara :

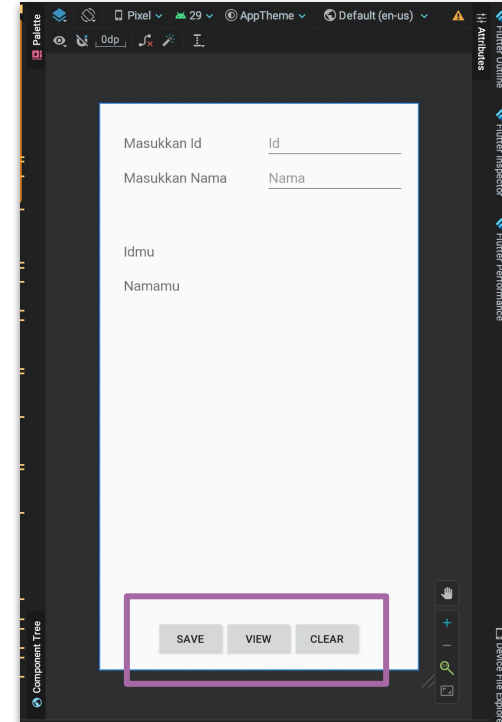
- Membuat Global variable nama SharedPreferences

```
private val sharedPrefFile = "kotlinsharedpreference"
```

- Membuat object sharedPreferences pada method onCreate():

```
private val sharedPrefFile = "kotlinsharedpreference"
```

Sipp banget deh! selanjutnya coba kamu perhatikan kotak warna ungu. Ada apa disitu?



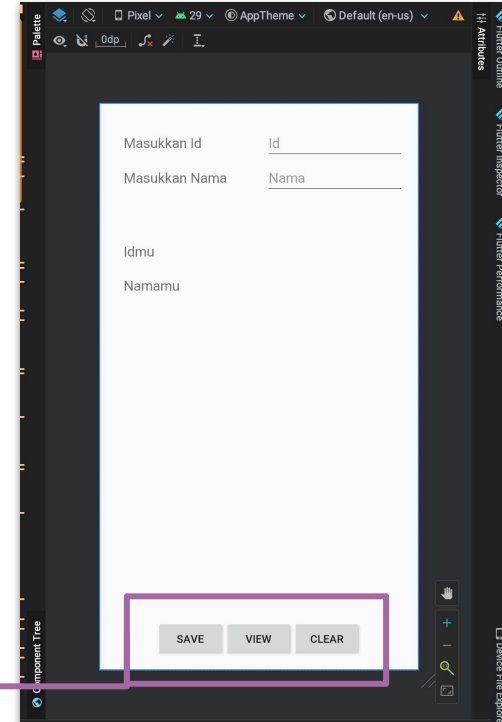


Iya betul. Ada button Save, button View dan Button Clear. Tadi kita udah membuat MainActivity.kt, sekarang kita fokuskan lagi pembahasan menuju tiga button cantik yang udah kamu jawab.

Kita coba bagi penjelasannya menjadi :

- **Konfigurasi tombol btnSave**
- **Konfigurasi tombol btnView**
- **Konfigurasi btnClear**

Tanpa lama lagi, langsung aja yuk buka tirai selanjutnya~





Konfigurasi Tombol btnSave

Pada MainActivity.kt, kita buat tombol btnSave untuk melakukan aksi menyimpan ID dan Nama yang ada pada EditText ke dalam SharedPreferences :

```
btnSave.setOnClickListener {  
    val id: Int = Integer.parseInt(etInputId.text.toString())  
    val name: String = etInputName.text.toString()  
    val editor: SharedPreferences.Editor = sharedPreferences.edit()  
    editor.putInt("id_key", id)  
    editor.putString("name_key", name)  
    editor.apply()  
    Toast.makeText(this@MainActivity, "Data Saved", Toast.LENGTH_SHORT).show()  
}
```



Konfigurasi Tombol btnView

Pada MainActivity.kt, kita buat tombol btnView untuk melakukan aksi pengambilan data ID dan name dari SharedPreferences.

Lalu, tampilkan pada tvShowName dan tvShowId

```
btnView.setOnClickListener {
    val sharedIdValue = sharedPreferences.getInt("id_key", 0)
    val sharedNameValue = sharedPreferences.getString("name_key", "defaultname")
    if (sharedIdValue.equals(0) && sharedNameValue.equals("defaultname")) {
        tvShowName.setText("default name: ${sharedNameValue}").toString()
        tvShowId.setText("default id: ${sharedIdValue.toString()}")
        Toast.makeText(this@MainActivity, "Data View Kosong", Toast.LENGTH_SHORT).show()
    } else {
        tvShowName.setText(sharedNameValue).toString()
        tvShowId.setText(sharedIdValue.toString())
        Toast.makeText(this@MainActivity, "Data View Ditampilkan", Toast.LENGTH_SHORT)
            .show()
    }
}
```



Konfigurasi Tombol btnClear

Pada MainActivity.kt, kita buat tombol btnClear untuk menghapus semua SharedPreferences yang tersimpan :

```
btnClear.setOnClickListener {  
    val editor = sharedPreferences.edit()  
    editor.clear()  
    editor.apply()  
    tvShowName.setText("")  
    tvShowId.setText("")  
    Toast.makeText(this@MainActivity, "Data Clear", Toast.LENGTH_SHORT).show()  
}
```




Kalau udah selesai, nanti wujud kodenya gimana?

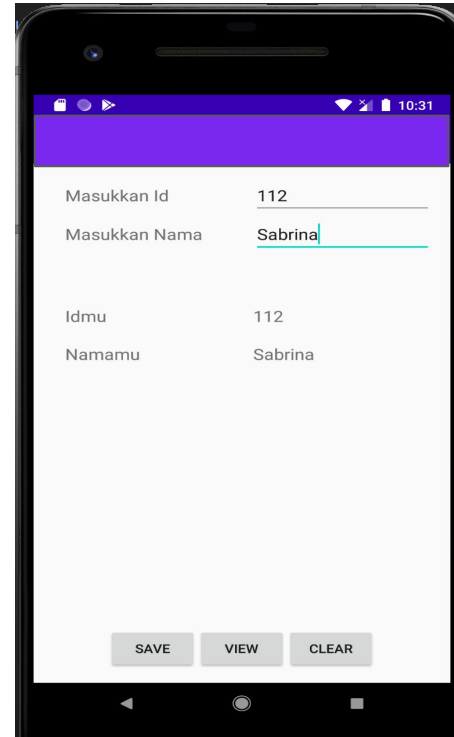
Bim salabim liat ke samping~

Secara keseluruhan, code yang dibuat pada MainActivity.kt adalah seperti ini 🙌

```
10 class MainActivity : AppCompatActivity() {
11     private val sharedPrefFile = "kotlinsharedpreference"
12     override fun onCreate(savedInstanceState: Bundle?) {
13         super.onCreate(savedInstanceState)
14         setContentView(R.layout.activity_main)
15         val sharedPreferences: SharedPreferences =
16             this.getSharedPreferences(sharedPrefFile, Context.MODE_PRIVATE)
17         btnSave.setOnClickListener { it: View?
18             val id: Int = Integer.parseInt(etInputId.text.toString())
19             val name: String = etInputName.text.toString()
20             val editor: SharedPreferences.Editor = sharedPreferences.edit()
21             editor.putInt("id_key", id)
22             editor.putString("name_key", name)
23             editor.apply()
24             Toast.makeText(context: this@MainActivity, text: "Data Saved", Toast.LENGTH_SHORT).show()
25         }
26         btnView.setOnClickListener { it: View?
27             val sharedIdValue = sharedPreferences.getInt("id_key", 0)
28             val sharedNameValue = sharedPreferences.getString("name_key", "defaultname")
29             if (sharedIdValue.equals(0) && sharedNameValue.equals( other: "defaultname")) {
30                 tvShowName.setText("default name: ${sharedNameValue}").toString()
31                 tvShowId.setText("default id: ${sharedIdValue}").toString()
32                 Toast.makeText(context: this@MainActivity, text: "Data View Kosong", Toast.LENGTH_SHORT).show()
33             } else {
34                 tvShowName.setText(sharedNameValue).toString()
35                 tvShowId.setText(sharedIdValue.toString())
36                 Toast.makeText(context: this@MainActivity, text: "Data View Ditampilkan", Toast.LENGTH_SHORT)
37                     .show()
38             }
39         }
40         btnClear.setOnClickListener { it: View?
41             val editor = sharedPreferences.edit()
42             editor.clear()
43             editor.apply()
44             tvShowName.setText("")
45             tvShowId.setText("")
46             Toast.makeText(context: this@MainActivity, text: "Data Clear", Toast.LENGTH_SHORT).show()
47         }
48     }
49 }
```



Tadi kan versi kodenya tuh, nah kalo versi tampilannya, bakal keliatan kayak gini nih 🙏





Biar makin paham, yuk ikut Challenge ini!

Buatlah Aplikasi sederhana yang terdiri dari SplashScreen, Layar Login, dan HomePage.

Pengguna bisa melakukan login, lalu status login tercatat di SharedPreferences.

Jadi, ketika pengguna keluar aplikasi lalu kembali lagi ke aplikasi, pengguna tidak perlu melakukan login lagi!

Pengumpulan tugas diserahkan kepada mentor

Kita akan membahas bersama-sama di pertemuan berikutnya.

Saatnya kita
Quiz!





1. Salah satu penyimpanan data lokal yang berupa XML tanpa memiliki struktur ...

- A. SharedPreferences
- B. MySQL
- C. SQLite



1. Salah satu penyimpanan data lokal yang berupa XML tanpa memiliki struktur ...

- A. SharedPreferences
- B. MySQL
- C. SQLite

Benar!. SharedPreferences adalah komponen salah satu penyimpanan lokal pada Aplikasi Android berupa file XML. Penyimpanan ini tidak memiliki struktur seperti layaknya Database (SQLite, MySQL, dll)



2. Untuk mendapatkan object SharedPreferences, method apakah yang bisa digunakan?

- A. `Context.getPreferences()`
- B. `getSharedPreferences()`
- C. `getSF()`



2. Untuk mendapatkan object SharedPreferences, method apakah yang bisa digunakan?

- A. `Context.getPreferences()`
- B. `getSharedPreferences()`
- C. `getSF()`

Benar! Kita bisa mendapatkan object SharedPreferences dengan menggunakan method `getSharedPreferences()`. Seperti contoh :

```
val sharedPreferences: SharedPreferences = this.getSharedPreferences(String
    preferences_fileName,int mode)
```




3. Untuk bisa mengedit SharedPreferences, kita membutuhkan object dari class SharedPreferences.Editor. Bagaimana kita mendapatkannya?

- A. Dengan method `edit()` dari object `SharedPreferences`
- B. Dengan method `editor()` dari object `SharedPreferences`
- C. Dengan method `editSharedPreferences()`



3. Untuk bisa mengedit SharedPreferences, kita membutuhkan object dari class SharedPreferences.Editor. Bagaimana kita mendapatkannya?

- A. Dengan method `edit()` dari object `SharedPreferences`
- B. Dengan method `editor()` dari object `SharedPreferences`
- C. Dengan method `editSharedPreferences()`

Benar! Dengan method `edit()` dari object `SharedPreferences`, kita bisa mendapatkan object dari class `SharedPreferences.Editor`

Seperti contoh :

```
val editor:SharedPreferences.Editor = sharedPreferences.edit()
```



4. Untuk menerapkan perubahan / edit terhadap SharedPreferences, kita bisa menggunakan method tertentu tanpa return type. Method tersebut adalah ...

- A. `editor.commit()`
- B. `editor.apply()`
- C. `editor.save()`



4. Untuk menerapkan perubahan / edit terhadap SharedPreferences, kita bisa menggunakan method tertentu tanpa return type. Method tersebut adalah ...

- A. `editor.commit()`
- B. `editor.apply()`
- C. `editor.save()`

Betul! Setelah kita melakukan pengeditan baik itu menambahkan, menghapus, mengubah suatu nilai di SharedPreferences, kita bisa menggunakan method `editor.apply()` untuk method yang tidak memiliki return type dan `editor.commit()` untuk method yang memiliki return type.



5. Terdapat di fitur manakah biasanya penerapan SharedPreferences?

- A. Login
- B. Alarm
- C. Notifikasi



5. Terdapat di fitur manakah biasanya penerapan SharedPreferences?

- A. Login
- B. Alarm
- C. Notifikasi

Betul! Fitur login adalah contoh dari penerapan SharedPreferences. Aplikasi akan mengingat bahwa user pernah melakukan login. Dengan demikian, ketika user mengakses aplikasi di lain waktu, user tidak perlu melakukan login lagi.



Referensi dan bacaan lebih lanjut~

1. [Android SharedPreferences in Kotlin](#)
2. [Android - Shared Preferences](#)
3. [kotlin android sharedpreferences - javatpoint](#)





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

Selanjutnya, kita bakal bahas metode pengaturan database, yaitu **Object Relational Mapping (ORM)**

Penasaran kayak gimana? Cus langsung ke topik selanjutnya~



Terima Kasih!



Next Topic

loading...