



Modul 3: Local Storage Data

Introduction



Jurusan Ilmu Komputer
Fakultas MIPA
Universitas Lampung



tutlane

dicoding



Google Developers
Authorized Training Partner



Introduction

- Pengembangan aplikasi pada platform manapun tidak luput dari proses penyimpanan dan manipulasi data.
- Pada perangkat Android, terdapat beragam mekanisme penyimpanan yang bergantung pada kebutuhan aplikasi.





Introduction

Pada umumnya penyimpanan dan pengolahan data di Android terbagi menjadi:

- 1. Penyimpanan Data pada Internal Storage**
- 2. Penyimpanan Data pada External Storage**
- 3. Penggunaan SharedPreferences**
- 4. SQLite Database**
- 5. Network**
- 6. Content Provider**





Penyimpanan Data pada Internal Storage

- Jenis penyimpanan ini menggunakan mekanisme pembuatan berkas yang secara umum dilakukan pada operasi berkas pemrograman berbasis Java.
- Setiap berkas yang tercipta akan secara otomatis tersimpan pada *internal storage* dan hanya bisa diakses oleh aplikasi yang membuatnya.
- Ketika pengguna mencopot aplikasi, maka semua berkas yang dibuatnya akan dihapus.





Penyimpanan Data pada Internal Storage

- Contoh membuat dan menulis pada berkas

```
1. String FILENAME = "hello_file";
2. String mString = "Hello Android";
3.
4. try {
5.     FileOutputStream fos = openFileOutput(FILENAME, MODE_PRIVATE);
6.     fos.write(mString.getBytes());
7.     fos.close();
8.     Log.d("WriteFile", "Success");
9. } catch (FileNotFoundException e) {
10.     e.printStackTrace();
11. } catch (IOException e) {
12.     e.printStackTrace();
13. }
```



Penyimpanan Data pada Internal Storage

- Sedangkan untuk membaca isi berkas:

```
1. try{
2.     FileInputStream fis = openFileInput(FILENAME);
3.     int c;
4.     String temp="";
5.     while( (c = fis.read()) != -1){
6.         temp = temp + Character.toString((char)c);
7.     }
8.
9.     tvText.setText("File contains "+temp);
10.
11. } catch (FileNotFoundException e) {
12.     e.printStackTrace();
13. } catch (IOException e) {
14.     e.printStackTrace();
15. }
```



Penyimpanan Data pada External Storage

- Saat ini hampir semua perangkat Android memiliki *external storage* dalam bentuk SD Card.
- Ia digunakan sebagai ruang penyimpanan tambahan.
- Karena media penyimpanan ini dapat dipasang dan dilepas dari perangkat Android, maka aplikasi harus memeriksa ketersediaannya terlebih dahulu.





Penyimpanan Data pada External Storage

```
1. /* Checks if external storage is available for read and write */
2. public boolean isExternalStorageWritable() {
3.     String state = Environment.getExternalStorageState();
4.     if (Environment.MEDIA_MOUNTED.equals(state)) {
5.         return true;
6.     }
7.     return false;
8. }
```




Penyimpanan Data pada External Storage

- Untuk menulis dan membuat berkas pada external storage maka aplikasi harus mendeklarasikan permission pada file AndroidManifest.xml seperti berikut:

```
1. <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```





Penggunaan SharedPreferences

- Media penyimpanan jenis ini akan memudahkan untuk menyimpan data dengan tipe data primitif seperti boolean, int, long, atau string yang tidak berelasi satu sama lain.
- Konsep yang digunakan pada jenis ini adalah pasangan antara *key-value*.
- Ini seperti implementasi *cookies* pada pengembangan aplikasi berbasis *web*.





Penggunaan SharedPreferences

- Contohnya pada sebuah kelas activity sebagai berikut:

```
1. public static final String PREFS_NAME = "MyPrefsFile";  
2.   
3. SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);  
4. SharedPreferences.Editor editor = settings.edit();  
5. editor.putBoolean("silentMode", mSilentMode);
```

- Pada kode di atas, silentMode adalah key dari mSilentMode.





Penggunaan SharedPreferences

- Ketika kita ingin mendapatkan nilai `silentMode`, mekanisme yang harus dilakukan adalah sebagai berikut :

```
1. SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);  
2. boolean silent = settings.getBoolean("silentMode", false);
```

- Dalam sebuah aplikasi Android, Anda bisa memiliki lebih dari satu data yang ingin disimpan.
- Kita dapat menggunakan banyak obyek `SharedPreferences` dengan nama yang berbeda-beda.





Penggunaan SharedPreferences

- Salah satu manfaat dari sharedPreferences adalah untuk menyimpan nilai-nilai konfigurasi yang ada di aplikasi.
- SharedPreferences akan selalu ada karena disimpan pada *memory*.
- Bahkan ketika aplikasi telah ditutup.





Penggunaan SharedPreferences

- Secara khusus, platform Android menyediakan activity khusus bernama `PreferenceActivity` dan `AppCompatActivity`
- Activity ini digunakan untuk memudahkan Anda membuat sebuah halaman konfigurasi dengan menggunakan `sharedpreferences`.





SQLite Database

- Ini adalah salah satu yang juara di Android.
- Anda bisa menyimpan data dalam bentuk tabel-tabel seperti yang dilakukan pada platform lain.
- Android menggunakan SQLite untuk memudahkan Anda mengelola semua data-data yang berelasi.
- Setiap *database* yang diciptakan dalam sebuah aplikasi hanya bisa diakses oleh aplikasi itu sendiri.





SQLite Database

- Mekanisme yang direkomendasikan untuk melakukan manipulasi data dengan SQLite adalah dengan menggunakan fasilitas SQLiteOpenHelper.
- Kelas ini merepresentasikan sebuah Data Definition Language (DDL) untuk membuat tabel-tabel yang diinginkan.





SQLite Database

- Contoh:

```
1. public class DictionaryOpenHelper extends SQLiteOpenHelper {  
2.     private static final int DATABASE_VERSION = 2;  
3.     private static final String DICTIONARY_TABLE_NAME = "dictionary";  
4.     private static final String DICTIONARY_TABLE_CREATE =  
5.         "CREATE TABLE " + DICTIONARY_TABLE_NAME + " (" +  
6.             KEY_WORD + " TEXT, " +  
7.             KEY_DEFINITION + " TEXT);";  
8.     DictionaryOpenHelper(Context context) {  
9.         super(context, DATABASE_NAME, null, DATABASE_VERSION);  
10.    }  
11.    @Override  
12.    public void onCreate(SQLiteDatabase db) {  
13.        db.execSQL(DICTIONARY_TABLE_CREATE);  
14.    }  
15. }
```



SQLite Database

- Untuk proses *query* pada *database*, kita bisa memanfaatkan metode `query()` yang menerima berbagai parameter pendukung seperti *selections*, *columns*, *grouping* dan lain sebagainya.
- Output dari hasil *query* adalah sebuah obyek Cursor.
- Obyek Cursor memudahkan kita mengakses hasil *query* dalam bentuk baris beserta kolomnya.





Network

- Sudah hal yang lumrah bagi aplikasi Android untuk mengakses data melalui jaringan Internet.
- Ada beragam protokol yang dapat digunakan, misalnya FTP, HTTP dan SMTP.
- Untuk HTTP sendiri, ada beberapa metode komunikasi yang didukung.
- Di antaranya post, get, put, delete dan head.





Network

- Hal ini mengapa disebut sebagai *Native Internet Based Application*.
- Ini adalah mekanisme penyimpanan terpusat yang memudahkan pengguna untuk melakukan proses pembuatan, pembacaan, pembaruan dan penghapusan data.





Network

- Untuk mendukung komunikasi data antara Android dengan penyedia data, Anda dapat membangun *web service*.
- Anda dapat mengembangkannya menggunakan bahasa PHP, Java, ASP, Python, Ruby atau Go.
- Pertukaran data antara aplikasi Android dengan *server* bisa menggunakan beragam pendekatan.





Network

- Anda dapat menggunakan *form-url-encoded*.
- Anda juga bisa menggunakan HTTPBody dengan bentuk obyek JSON.
- Kita sudah pernah memproses data JSON pada modul sebelumnya, yaitu ketika Anda membuat aplikasi yang menampilkan data dari OpenWeatherMap.





Content Provider

- Sebetulnya content provider bukanlah sebuah mekanisme penyimpanan di Android.
- Ia merupakan sebuah antarmuka untuk menjadikan aplikasi sebagai penyedia data bagi aplikasi lain di dalam sebuah peranti.
- Dengan mekanisme di atas, setiap aplikasi dapat memberikan akses kepada aplikasi lain untuk menggunakan data yang dimilikinya.





Content Provider

- Data yang dapat digunakan oleh content provider bisa disimpan di dalam *database* SQLite, berkas ataupun data yang berada pada *server* tertentu.
- Sebagai contoh, aplikasi *chatting* dan aplikasi kontak yang terdapat di Android menggunakan sumber data yang sama untuk melakukan manipulasi data kontak.





CodeLab