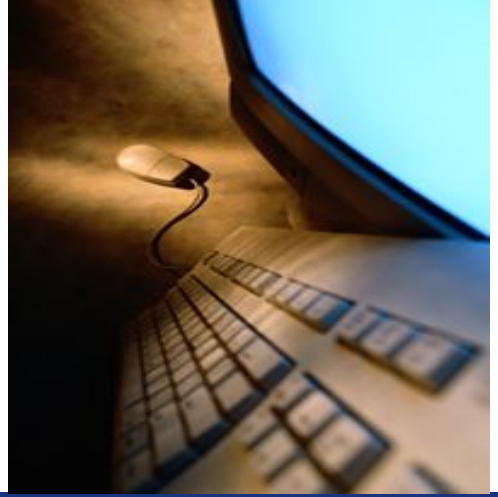
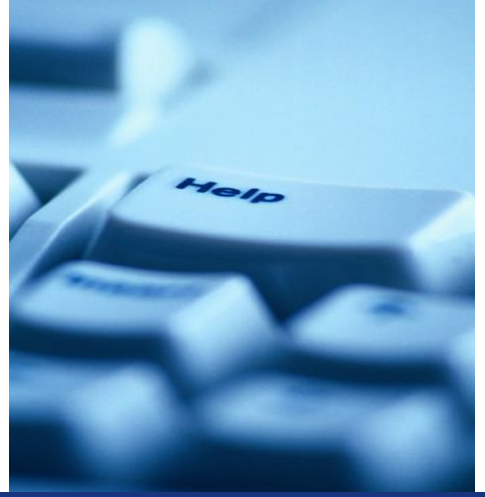
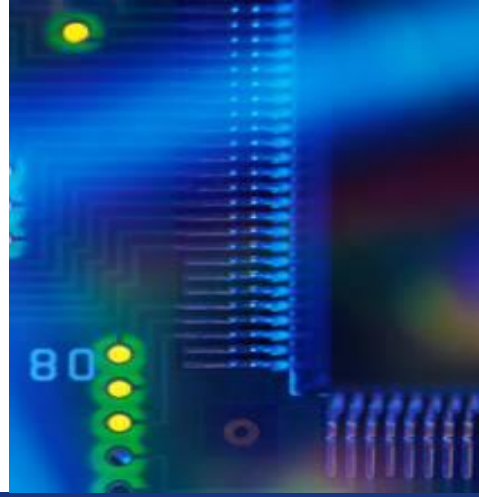
UNIVERSITAS
BUDI LUHUR



FAKULTAS TEKNOLOGI INFORMASI

Mobile Programming

[PG119/ 3 SKS]

Riskiana Wulan, MKom

DATA Storage and Persistence in Android


Why Data Storage Matters in Mobile Apps

Importance:

- Data persistence is crucial for saving user information, app settings, and content between sessions.
- Proper storage ensures smooth user experiences and app reliability.

Types of Data to Store:

- **Small Data**: Preferences, settings.
- **Structured Data**: Records like user profiles, transactions.
- **Files**: Images, PDFs, etc.


Android Data Storage Options

1. **SharedPreferences**
   - Key-value pairs.
   - Ideal for simple configurations
   - e.g., Saving user preferences like theme or login status).
2. **Internal Storage**
   - Store private files in the device's internal memory.
   - Data is sandboxed and secure.
   - e.g Saving app-specific files like text notes securely
3. **External Storage**
   - Store large files (e.g., images, videos).
   - Requires permissions; less secure.
   - Shared files accessible by other apps.

4. SQLite Database

- Lightweight relational database for structured data.
- Structured data with SQL support.
- Ideal for apps requiring complex queries.

5. Room Persistence Library

- Abstraction layer over SQLite.
- Reduces boilerplate code and integrates with LiveData.

https://developer.android.com/training/data-storage


WHY SQLite

- **Advantages:**
  - Lightweight and embedded directly in Android.
  - Ideal for relational data storage.
  - Supports complex queries using SQL.
- **Use Cases:**
  - Apps requiring user profiles, task management, or offline data.
  - Any app needing local structured storage.

Steps to Use SQLite

Create a Database and Table

- Define the schema in an SQLite helper class.

Insert Data

- Use ContentValues and db.insert().

Read Data

- Use Cursor to fetch data.

Update Data

- Update records with db.update().

Delete Data

- Remove records using db.delete().

What is SQLite?

Definition:
SQLite is an embedded relational database available in Android by default.

Key Features:

- Lightweight and serverless.
- Uses SQL for queries.
- Ideal for structured local storage.

Challenges with SQLite

- No compile-time verification of SQL queries.
- Manual boilerplate code for CRUD operations.
- No direct support for observability (e.g., LiveData).

Sample of creating an SQLite database in Android

```
SQLiteDatabase db = openOrCreateDatabase("app_db", MODE_PRIVATE, null);
db.execSQL("CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY, name TEXT, email TEXT)");
db.execSQL("INSERT INTO users (name, email) VALUES ('John Doe', 'john.doe@example.com')");
```

https://www.sqlite.org/

WHat is Room Datastore?

Definition: Room is a persistence library part of Android Jetpack that provides an abstraction layer over SQLite.

Benefits:

- Simplifies database interaction.
- Provides compile-time verification for SQL queries.
- Supports LiveData and Flow for observable queries.

https://developer.android.com/training/data-storage/room

Room Architecture

Main Components:

1. **Entity**: Represents a table in the database.
2. **DAO (Data Access Object)**: Contains methods to access the database.
3. **Database**: Serves as the main access point.

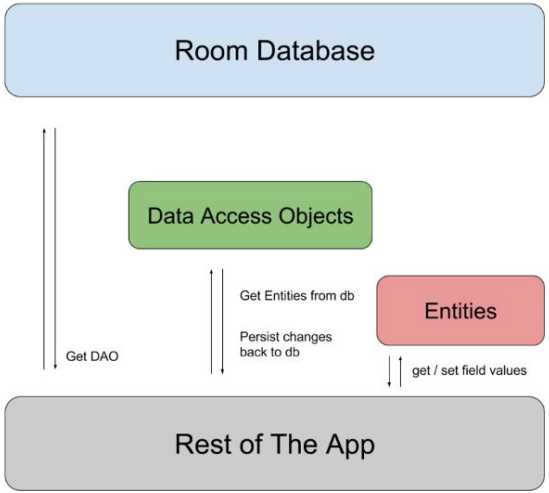


Figure 1. Diagram of Room library architecture.

Why Use Room Over SQLite?

Room Benefits:

- Simplifies SQL usage with annotations.
- Compile-time SQL verification.
- Supports LiveData, Flow, and Kotlin Coroutines.
- Built-in migrations.

ROOM IMPLEMENTATION


Room Implementation

1. Adding room dependencies (in build.gradle)

```
implementation 'androidx.room:room-runtime:2.5.0'
annotationProcessor 'androidx.room:room-compiler:2.5.0'
```

2. Create an Entity class (User.java)

```
@Entity(tableName = "users")
public class User {
    @PrimaryKey(autoGenerate = true)
    public int id;

    @ColumnInfo(name = "name")
    public String name;

    @ColumnInfo(name = "email")
    public String email;
}
```

Notes:

- **@Entity**: Declares the table.
- **@PrimaryKey**: Marks the primary key.
- **@ColumnInfo**: Customizes column names.

3. Create a DAO class (UserDao.java)

```
@Dao
public interface UserDao {
    @Insert
    void insertUser(User user);

    @Query("SELECT * FROM users")
    List<User> getAllUsers();

    @Delete
    void deleteUser(User user);
}
```

@Insert, @Query, @Delete: Predefined annotations for database operations.

4. Create the database (AppDatabase.java)

```
@Database(entities = {User.class}, version = 1)
public abstract class AppDatabase extends
RoomDatabase {
   public abstract UserDao userDao();
}
```

Note:

Masukkan entity class di dalam entities database


Trigger create database from MainActivity

```java
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //create database
        AppDatabase db = Room.databaseBuilder(getApplicationContext(),
                AppDatabase.class,  name: "user-database").build();

        //insert
        User user = new User();
        user.name = "John Doe";
        user.email = "john.doe@example.com";

        new Thread(() -> db.userDao().insertUser(user)).start();

        //query
        new Thread(() -> {
            List<User> users = db.userDao().getAllUsers();
            for (User userData : users) {
                Log.d( tag: "User", userData.name);
            }
        }).start();


    }
}
```

Lalu kita bisa melakukan testing dengan run app melalui emulator

Note:

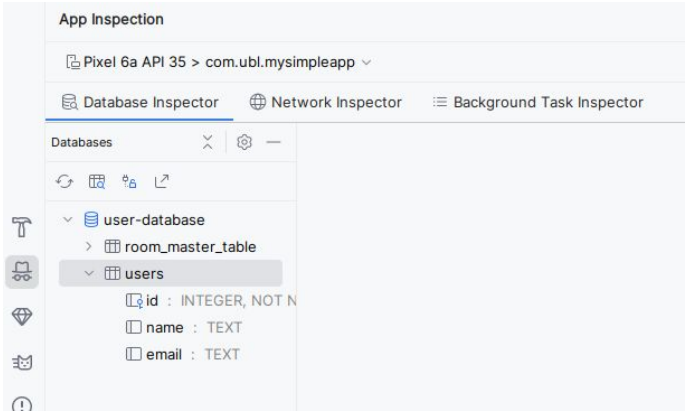
Use a separate thread for database operations to avoid blocking the UI thread.

Debug database in Android

Pilih view->tool window->App inspection

Perhatikan di tab Database

Inspector, terdapat user-database beserta table users yang berhasil terbentuk



Kita dapat melakukan query langsung ke database melalui database inspector

HANDS ON LAB


Lanjutan Student Registration form

Buat database untuk menyimpan student registration dengan Room

- **Pada saat button register diklick, maka new student akan tersimpan ke database**
- **Tampilkan pada summary activity berupa record student yang berasal dari database**