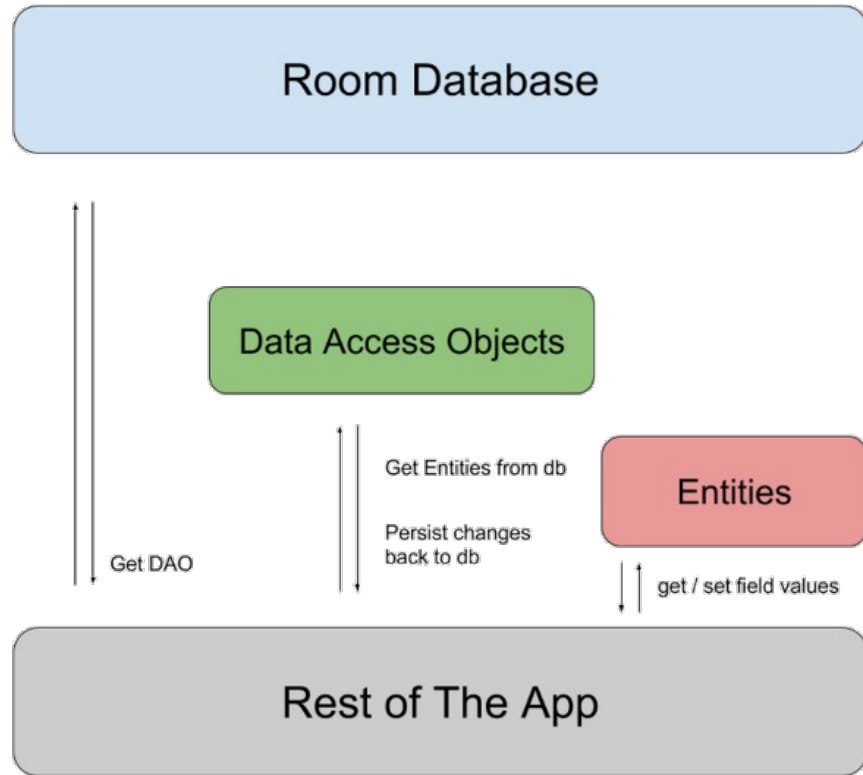


ROOM Database

Pengembangan Aplikasi Perangkat Bergerak

Apa itu Room?

- **Room merupakan presistence library bagian dari Android Jetpack**
- **Room memudahkan kita mengakses database SQLite dari pada menggunakan SQLiteDatabase**
- **Keunggulan Room:**
 - **Sudah Mendukung proses validasi pada proses compile**
 - **Mudah diintegrasikan dengan arsitektur android lainnya seperti LiveData dan RXJava**



Room terdiri dari 3 komponen:

- Entity
- DAO
- Database

Entity

- Mewakili tabel dalam database
- Class diberikan annotation `@Entity(tableName="nama tabel")`
- Atribut diberikan annotation `@ColumnInfo(name = "nama kolom")`
- Atribut yang menjadi primary key diberi annotation `@PrimaryKey(autoGenerate = "true/false")`

```
@Entity(tableName = "item")
public class Item {
    @PrimaryKey(autoGenerate = true)
    private int id;
    @ColumnInfo(name = "judul")
    private String judul;

    //setter dan getter.....
}
```

DAO

- **DAO** adalah class yang bertanggungjawab untuk operasi pada database, seperti proses **select, insert, update, delete**
- **Class** diberikan annotation **@Dao**

```
@Dao
public interface ItemDao {
    @Query("SELECT * FROM item")
    List<Item> getAll();

    @Insert
    void insertAll(Item... items);

    @Delete
    void delete(Item item);

    @Update
    void update(Item item);
}
```

Database

- **Class yang mewakili koneksi ke database**
- **Abstract Class turunan dari class RoomDatabase**
- **Class diberikan annotation @Database dan memuat daftar nama class entity**
- **Memiliki abstract method tanpa argument dan tipe kembalian adalah class Dao**
- **Hanya boleh ada satu objek instansiasi class Database pada satu waktu untuk menghindari proses update pada database yang saling mengganggu.**

Database

```
@Database(entities = {Item.class}, version = 1)
public abstract class AppDatabase extends RoomDatabase {

    .....

    public abstract ItemDao itemDao();

    public static AppDatabase getInstance(Context context) {
        if (sInstance == null) {
            synchronized (LOCK) {
                Log.d(LOG_TAG, "Creating new database instance");
                sInstance = Room.databaseBuilder(context.getApplicationContext(),
                    AppDatabase.class, AppDatabase.DATABASE_NAME)
                        .build();
            }
        }
        Log.d(LOG_TAG, "Getting the database instance");
        return sInstance;
    }
}
```

Executor

- **Selalu eksekusi method dari Dao class diluar Uithread**
- **Executor merupakan salah satu cara membuat thread**

```
public static AppExecutors getInstance() {  
    if (sInstance == null) {  
        synchronized (LOCK) {  
            sInstance = new AppExecutors(Executors.newSingleThreadExecutor(),  
                Executors.newFixedThreadPool(3),  
                new MainThreadExecutor());  
        }  
    }  
    return sInstance;  
}
```


Menggunakan Room dalam activity

```
@Override
public void onClick(View view) {
    i++;
    Item item = new Item();
    item.setId(i);
    item.setJudul(et1.getText().toString());
    AppExecutors.getInstance().diskIO().execute(new Runnable() {
        @Override
        public void run() {
            appDb.itemDao().insertAll(item);
        }
    });
}
```

Kode sumber contoh bisa diakses di
<https://gitlab.com/p989/java/roomexample1>