JURIKOM (Jurnal Riset Komputer), Vol. 7 No. 3, Juni 2020
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v7i3.2161
Hal 376-381

Perbandingan Antara SQLite, Room, dan RBDLiTe dalam Pembuatan Basis Data pada Aplikasi Android

Raden Bagus Dimas Putra, Eko Setia Budi, Abdul Rahman Kadafi

Program Studi Sistem Informasi, STMIK Nusa Mandiri, Jakarta, Indonesia
Email: [1]raden.rbd@nusamandiri.co.id, [2]eko.etb@nusamandiri.co.id, [3]abdul.alk@nusamandiri.co.id
Submitted 05-05-2020; Accepted 29-05-2020; Published 14-06-2020

Abstrak

Android adalah salah satu platform sistem operasi seluler terbesar saat ini. Di tengah terus meningkatnya kebutuhan pengguna, programmer android masih kurang untuk memenuhi kebutuhan pasar saat ini. Salah satu masalah yang menyebabkan ini adalah kesulitan membuat database internal sehingga banyak orang menyerah untuk mempelajari membuat aplikasi android. Basis data internal android sulit dibuat karena seluruh struktur data pada basis data tersebut harus dibangun dengan query yang dijalankan dalam kode program yang dibuat. Beberapa basis data internal yang umum digunakan saat ini adalah SQLiTe dan Room. Kedua basis data tersebut cukup sulit dipelajari dimana SQLite harus membangun semuanya dari awal sedangkan Room harus mempelajari struktur dan sistematika penulisan library tersebut. Oleh karena itu pada tulisan ini, penulis mengusulkan suatu pustaka serta template yang diberi nama "Relational Basis Data Library dan Template" yang nantinya akan disebut sebagai RBDLiTe yang dapat membuat database internal untuk aplikasi Android dengan mudah yang juga akan dibandingkan dengan SQLiTe dan Room dalam penggunaaanya.

Kata Kunci: Basisdata, Android, SQLite, Room, RBDLiTe

Abstract

Android is one of the largest mobile operating system platforms today. Amid the ever-increasing needs of users, Android programmers are still lacking to meet the current market needs. One of the problems that causes this is the difficulty of creating an internal database, so many people give up to learn to make Android applications. Android's internal database is hard to create because the entire data structure on that database should be built with the query executed in the created program code. Some of the common internal databases currently used are SQLiTe and Room. Both databases are quite difficult to learn where SQLite should build everything from scratch while Room should study the structure and systematics of of the library. Therefore, in this article, the author proposes library and template named "Relational Database Library and Template" which will be referred to as RBDLiTe which can create an internal database for Android applications easily which will also be compared with SQLiTe and Room in its use.

Keywords: Database, Android, SQLite, Room, RBDLiTe

1. PENDAHULUAN

Android adalah salah satu platform sistem operasi seluler terbesar saat ini. Di tengah meningkatnya kebutuhan penggunaan, programmer Android masih kurang untuk memenuhi kebutuhan pasar saat ini. Salah satu masalah yang menyebabkan ini adalah kesulitan membuat database internal. Beberapa orang menyerah untuk terus menjadi seorang programmer Android karena tersebut. Untuk pembuatan basis data internal, pemrogram perlu menafsirkan kueri DDL atau DML secara manual. Selain itu, banyak jenis kelas dan objek yang digunakan, sehingga untuk memahami pembuatan proses terlihat sangat sulit. Selain itu basis data internal android sulit dibuat karena seluruh struktur data pada basis data tersebut harus dibangun dengan query yang dijalankan dalam kode program yang dibuat. Beberapa basis data internal yang umum digunakan saat ini adalah SQLiTe dan Room.

SQLite adalah basis data open source yang tertanam ke dalam Android. SQLite mendukung fitur database relasional standar seperti sintaks SQL, fungsi transaksi dan jugan fungsi *prepared statement*. Selain itu, hanya membutuhkan sedikit memori pada saat dijalankan. SQLite adalah perpustakaan perangkat lunak yang menerapkan sifat mandiri, tanpa server, tanpa konfigurasi, dan mesin transaksional basis data SQL. SQLite adalah mesin database SQL yang paling banyak digunakan di dunia [1]. SQLite mendukung jenis data teks (mirip dengan string di Java), BILANGAN bulat (mirip dengan panjang di Java) dan REAL (mirip dengan ganda di Java). Semua jenis lainnya harus dikonversi menjadi salah satu bidang ini sebelum menyimpannya dalam database. SQLite sendiri tidak memvalidasi jika jenis ditulis ke kolom sebenarnya dari jenis yang didefinisikan, misalnya menulis integer ke dalam kolom string dan sebaliknya. Pernyataan SQL untuk membuat dan memperbarui database harus hanya didefinisikan dan database secara otomatis dikelola untuk pengguna oleh platform Android. Berdasarkan penjelasan tersebut penggunaan SQLite masih terasa sulit karena programmer harus membangun semuanya sendiri karena tanpa server dan juga mengkonfigurasi sendiri.

Basis data Room menyediakan antarmuka untuk database yang didasari oleh SQLite. Room juga menyediakan repositori ke objek akses data (DAO). Sebuah aplikasi hanya harus memiliki satu instance database Room yang kemudian dapat digunakan untuk mengakses beberapa tabel database[2]. DAO berisi pernyataan SQL yang diperlukan oleh repositori untuk memasukkan, mengubah, mengambil dan menghapus data dalam database SQLite. Pernyataan SQL ini dipetakan ke metode yang kemudian dipanggil dari dalam repositori untuk mengeksekusi query yang sesuai. Entitas adalah kelas yang mendefinisikan skema untuk tabel dalam database serta mendefinisikan nama tabel, nama kolom, jenis data, dan mengidentifikasi kolom yang akan menjadi primary key. Selain mendeklarasikan skema tabel, kelas entitas juga berisi metode pengambil dan penyetel yang menyediakan akses ke bidang data ini. Data dikembalikan ke repositori oleh DAO dalam menanggapi panggilan metode query SQL yang akan mengambil bentuk objek dari kelas entitas ini. Metode *Getter* kemudian akan dipanggil untuk mengekstrak data dari objek entity. Demikian pula, ketika repositori perlu menulis data baru

JURIKOM (Jurnal Riset Komputer), Vol. 7 No. 3, Juni 2020
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v7i3.2161
Hal 376-381

ke database, itu akan membuat objek entitas, kemudian mengkonfigurasi nilai pada objek melalui metode *Setter*, kemudian memanggil metode insert yang dideklarasikan di DAO yang membawa objek entitas untuk disimpan. Database Room menggunakan SQLite yang bertanggung jawab untuk menyimpan dan menyediakan akses ke data. Kode aplikasi, termasuk repositori, tidak boleh membuat akses langsung ke database. Semua operasi database dilakukan dengan menggunakan kombinasi database ruang, DAOs dan entitas. Dengan menggunakan database Room pengguna sudah memiliki kerangka kerja namun banyak aturan-aturan yang sulit dan harus dipelajari seperti mempelajari kerangka kerja baru. Penggunaan kerangka kerja pada umumnya digunakan oleh programmer yang cukup memiliki pengalaman, dimana jika baru mepelajari suatu Bahasa pemrograman dan langsung menggunakan kerangka kerja akan terasa sangat sulit.

Berdasarkan [3]–[5], template telah digunakan untuk memfasilitasi pemahaman konseptual tentang konsep pemrograman. Yuen dan Liu [4] mempelajari bagaimana template permainan video didukung dan pemahaman konseptual dipandu pemrograman berorientasi objek (PBO) yang kemudian digunakan siswa untuk membuat permainan multimedia interaktif mereka sendiri. Dalam menggunakan template, Yuen dan Liu menemukan tiga tingkat interaksi yang menghasilkan pemahaman konseptual yang lebih baik. Pertama, umpan balik visual dari permainan, khususnya perilaku tak terduga, membawa perhatian pada kekurangan dalam desain perangkat lunak. Ketika siswa mengalami kesalahan ini, mereka mulai tingkat lain interaksi dengan template. Kedua adalah untuk mengeksplorasi sumber daya template. Eksplorasi ini dapat mengarah pada penemuan informasi penting, penemuan hubungan antara sumber daya, dan kode model yang nantinya dapat digunakan. Ketiga, ketika mereka menjelajahi, siswa mulai memperbaiki kode mereka serta pemahaman mereka tentang kode. Pada tingkat kognitif, template meningkatkan pemahaman konseptual karena menyediakan dasar atau kode dasar yang tidak lengkap dari mana siswa dapat membangun program mereka. Jika dirancang dengan benar, kode template dapat digunakan sebagai model atau teladan yang memandu pemikiran dan pengkodean siswa[6]. Berdasarkan masalah ini penulis membuat pustaka untuk menghasilkan *query* sehingga tidak perlu membangun database menggunakan query karena bisa di generate, serta template kelas untuk pembuatan kelas database, sehingga penciptaan database internal akan mudah, cukup klik dan input data sesuai kebutuhan. Pustaka dan template tersebut kemudian diberi nama RBDLiTe

2. METODE PENELITIAN

Pada tulisan ini, tahap penulis melukakan penelitian yaitu: merumuskan masalah, mencari teori pendukung terkait, membuat RBDLiTe, Membuat tiga program serupa dengan tiga metode berbeda, menganalisis dan membandingkan kode program, serta menarik kesimpulan dari hasil analisa. Adapun alur penelitian dapat juga dilihat pada Gambar 1.

Gambar 1. Diagram aktivitas alur penelitian

a. Merumuskan Masalah
   Pada tahap ini, penulis mencari permasalahan mengenai pembuatan basis data internal pada aplikasi android serta merumuskan solusi yang memungkinkan.
b. Mencari Teori Pendukung
   Pada tahap ini, penulis mempelajari secara dalam metode umum yang digunakan untuk membuat basis data internal, serta mencari teori-teori bagaimana menguraikan kerumitan kode program sesuai dengan hasil rumusan masalah.
c. Membangun RBDLiTe
   Bedasarkan teori pendukung, dibuatlah RBDLiTe untuk memecahkan masalah yang dirumuskan.
d. Membuat Program dengan SQLite, Room, dan RBDLiTe
   Pada tahap ini, penulis membuat tiga program yang sama serta fungsi yang sama dengan basis data SQLite, Room, maupun RBDLiTe untuk kemudian dibandingkan. Aplikasi yang dibandingkan akan dibuat menggunakan Editor Android Strudio dan dijalankan menggunakan handphone dengan OS Android versi 9.
e. Menganalisis dan Membandingkan Kode Program
   Dari hasil program yang dibuat kemudian dianalisis untuk mengetahui kelebihan dan kekurangan dari masing-masing basis data. Setelah menganalisis kemudian melakukan testing untuk aplikasi yang dibuat dan memeriksa apakah hasilnya adalah sama. Hasil akhir harus sama untuk memberikan bobot yang adil dalam penilaian pembuatan basis data.
f. Menarik Kesimpulan

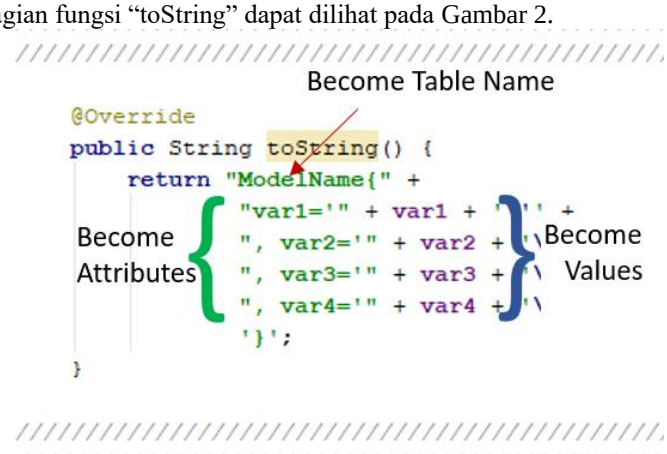
JURIKOM (Jurnal Riset Komputer), Vol. 7 No. 3, Juni 2020
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v7i3.2161
Hal 376-381

Dari hasil analisis dan perbandingan kemudian dibuat kesimpulan apakah RBDLiTe dapat menyelesaikan rumusan masalah yang ada.

3. HASIL DAN PEMBAHASAN

3.1 Pembuatan *Library* pada RBDLiTe

RBDLiTe *Library* yang kemudian diberi nama RBDLib, dibuat berdasarkan pada hasil yang dihasilkan fungsi yang disebut "toString" saat pembentukan model. Fungsi ini merupakan bagian dari "*Plain Old Java Objects*" yang lebih dikenal dengan nama POJO. Pada Sebagian besar aplikasi editor sudah dapat me-*generate* fungsi ini, sehingga kode untuk fungsi tersebut tidak perlu diketik. Hasil dari fungsi ini kemudian dibagi menjadi beberapa bagian dan diberi label sebagai nama tabel, atribut, dan nilai. Hasil pembagian fungsi "toString" dapat dilihat pada Gambar 2.



Gambar 2. Pembagian fungsi "toString" menjadi beberapa data

SQL query dibuat berdasarkan hasil pembagian pada Gambar 2. Dalam pembuatan query, penulis menggunakan kelas StringBuilder yang objeknya diberi nama "sb", nama tabel dengan nama "tbName". Untuk atribut dan nilai yang dibuat menggunakan kelas ArrayList objek yang bernama "attributes" dan "values". Selain itu, penulis menggunakan variabel "i" yang merupakan bilangan bulat untuk perulangan dalam proses pengambilan data pada "attributes" dan "values". Semua objek dan variabel yang digunakan dibuat dalam bentuk struktur data global dengan visibilitas pribadi. Contoh pembuatan query insert dapat dilihat pada Gambar 3.

```
public String toInsert(){
    sb.setLength(0);
    sb.append("INSERT INTO ");
    sb.append(tbName);
    sb.append("(");
    for (i=0; i<attibutes.size(); i++){
        sb.append(attibutes.get(i));
        if(i<attibutes.size()-1)
            sb.append(",");
    }
    sb.append(") VALUES(");
    for (i=0; i<values.size(); i++){
        sb.append(values.get(i));
        if(i<values.size()-1)
            sb.append(",");
    }
    sb.append(");");
    return sb.toString();
}
```

Gambar 3. Contoh pembuatan query insert

Umumnya, query SQL dibagi menjadi dua bagian, yaitu *data definition language* atau DDL dan *data manipulation language* atau DML. Dalam RBDLiTe ada beberapa fungsi yang dibangun di antara mereka adalah:

a. DDL
   1. *Create Table*
   2. *Drop Table*
   3. *Alter Table*

JURIKOM (Jurnal Riset Komputer), Vol. 7 No. 3, Juni 2020
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v7i3.2161
Hal 376-381

b. DML
   1. *Select*
   2. *Insert*
   3. *Update*
   4. *Delete*

Beberapa fungsi yang ada di *library* dibuat menggunakan metode *overloading*, sehingga satu fungsi dapat memiliki berbagai jenis input dan hasil. Penggunaan *polimorfisme overloading* pada fungsi bertujuan untuk memenuhi beragam kebutuhan yang dapat digunakan dengan nama yang sama seperti menggunakan klausa "where" atau tanpa klausa "where" pada query yang dibuat.

3.2 Pembuatan *Template* pada RBDLiTe

Template dibuat menggunakan kelas turunan dari SQLiteOpenHelper yang pada dasarnya sudah memiliki atribut dan fungsi dari kelas supernya. Kelas ini kemudian diintegrasikan dengan membuat objek dari kelas RBDLib sehingga template tidak akan dapat digunakan tanpa menerapkan RBDLib. Dengan bantuan sebuah objek dari RBDLib, isi dari template akan terlihat lebih sederhana karena sebagian besar kompleksitas sudah dilakukan oleh RBDLib. Untuk pembuatan database, penulis membuat modifikasi pada fungsi super yang diantaranya:

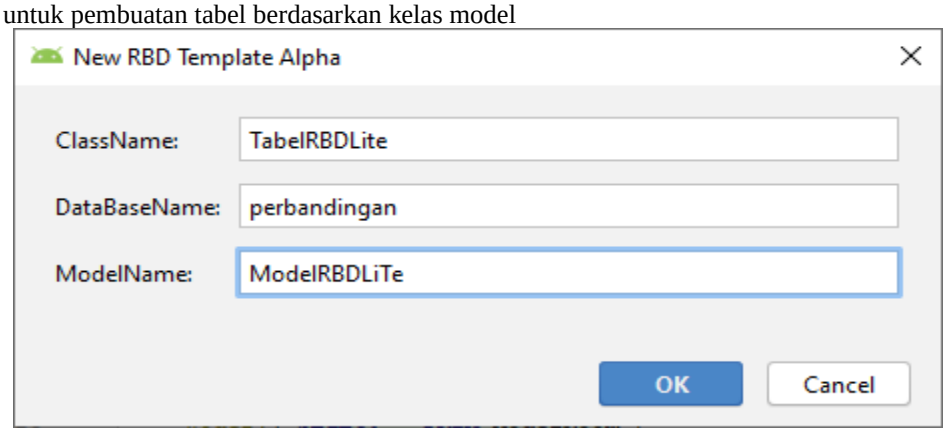
a. Constructor: penambahan parameter "*source*" sebagai dasar pembentukan tabel berdasarkan objek yang pengguna memberikan.
b. "onCreate" ditambahkan fungsi untuk membuat tabel database
c. "onUpgrade" penambahan fungsi untuk mengganti tabel lama dengan tabel baru

Selain mengubah fungsi bawaan, untuk memfasilitasi kebutuhan pengguna, penulis juga menambahkan fungsi dasar untuk mengelola database yang diantaranya adalah:

a. "getData" digunakan untuk mengakses database menggunakan objek dari kelas "Cursor" atau pointer yang menunjuk ke basis data.
b. "getDataArray" digunakan untuk mendapatkan tabel data pada basis data dalam bentuk ArrayList dua dimensi.
c. "insertData" digunakan untuk menambahkan data ke tabel di basis data.
d. "updateData" digunakan untuk mengubah data pada tabel dalam basis data.
e. "deleteData" digunakan untuk menghapus data pada tabel dalam basis data

Fungsi yang dibuat penulis hanya memeuhi kebutuhan standar pengolahan data dimana kode hasil template masih dapat dirubah oleh pengguna untuk memenuhi kebutuhan lainnya. Untuk melakukan kegiatan umum pengguna tidak perlu lagi membuat kode programnya cukup memasukan inputan yang diminta, lalu file kelas database siap digunakan. Adapun inputan yang diperlukan untuk membentuk fila kelas database dalah:

a. ClassName: adalah nama kelas yang dibuat.
b. DataBaseName: untuk nama database yang digunakan
c. ModelName: untuk pembuatan tabel berdasarkan kelas model



Gambar 4. Contoh pembuatan kelas tabel pada database menggunakan template

3.3 Perbandingan Instalasi SQLite, Room, dan RBDLiTe

a. SQLite

Pada SQLite pengguna tidak perlu menginstall atau mengiplementasikan apapun karena SQLite adalah kelas dasar yang merupakan bawaan dari pemrograman android.

b. Room

Untuk menggunakan Room pengguna harus mengimplementasikan *library* Room pada bagian *dependencies* yang ada pada *gradle* seperti pada Gambar 5.

```
implementation 'androidx.room:room-runtime:2.2.5'
annotationProcessor 'androidx.room:room-compiler:2.2.5'
```

Gambar 5. Contoh kode pengimplementasian Room

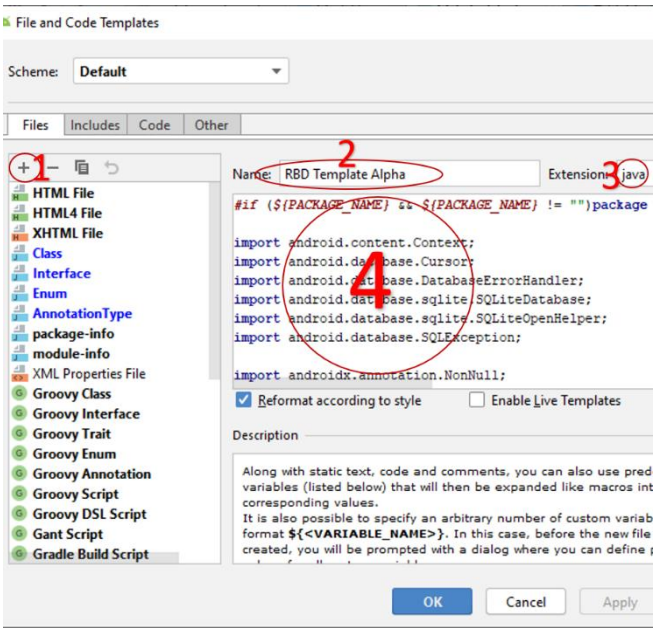
JURIKOM (Jurnal Riset Komputer), Vol. 7 No. 3, Juni 2020
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v7i3.2161
Hal 376-381

c. RBDLiTe

Untuk menggunakan RBDLiTe pengguna harus mengunduh file pada tautan http://bit.ly/2I6vPDG. Setelah itu mengsektraknya. Lalu pindahkan "RBDLibrary_Alpha" ke folder "libs" yang ada pada folder *project* yang dibuat lalu ke folder "app". Setelah disalin ke folder libs buatlah implementasi pada bagian *dependencies* yang ada pada *gradle* seperti pada Gambar 6. Untuk menerapkan template di Android Studio editor, pengguna dapat memilih "file" kemudian "new" kemudian "Edit file template...". Setelah kotak dialog pembuatan template terbuka lakukan langkah seperti pada keterangan Gambar 7.

```
implementation files('libs\\RBDLibrary_Alpha.jar')
```

Gambar 6. Contoh kode pengimplementasian RBDLibrary



Gambar 7. Contoh dialog box pembuatan template

1. Pilih *button* yang diberi label "1"
2. Isi kotak "2" dengan nama template
3. Pastikan kotak "3" berisi "java"
4. Salin isi RBDTemplate.txt ke dalam box "4"

3.4 Perbandingan Penggunaan SQLite, Room, dan RBDLiTe

a. SQLite

Pada SQLite, semua query harus dibuat sendiri, harus membuat data akses objek dengan proses dibentuk sendiri, serta membuat database dan tamble sendiri. Semua query dan arsitektur pembuatan database harus dipelajari dan dibuat sendiri sehingga kode akan terlihat panjang, rumit, dan sulit dipelajari. Untuk contoh perbandingan model pada SQLite, Room, dan RBDLite dapat dilihat pada Gambar 8.

b. Room

Walaupun merupakan turunan dari SQLiTe, pada penggunaan basis data Room, Hampir semua kerangka pembuatan basis data digantikan dengan "*annotation*" seperti pada Gambar 8 pada bagian "ModelRoom". Pengguna tidak perlu mempelajari kerangka basis data SQLite kembali, cukup mempelajari kerangka kerja Room. Jangkauan jumlah fungsi yang dapat digunakan sangat banyak sehingga dapat memenuhi hampir seluruh kebutuhan pengguna. Karena cakupan sangat luas, mempelajari Room sendiri butuh pengalaman yang cukup walaupun terlihat sangat simple dengan jumlah kode yang pendek. Sangat cocok untuk programmer yang sudah cukup pengalaman.

JURIKOM (Jurnal Riset Komputer), Vol. 7 No. 3, Juni 2020
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v7i3.2161
Hal 376-381

```java
package com.rbd.roomvsrbdlite.model;

public class ModelSQLite {
    String var1,var2,var3,var4;

    /* Generate Constructor, Getter dan Setter sama
       antara SQLite, Room, Maupun RBDLiTe */

    public String toCreate(){
        return "CREATE TABLE ModelSQLite ( " +
                "var1 TEXT " +
                ", var2 TEXT " +
                ", var3 TEXT " +
                ", var4 TEXT " +
                " )";
    }

    public String toInsert(){
        return "INSERT INTO ModelSQLite VALUES ("
                + "'" + var1 + "'"
                + ",'" + var2 + "'"
                + ",'" + var3 + "'"
                + ",'" + var4 + "'"
                + ")";
    }

    public String toUpdate(){...}

    public String toDelete(){...}

    public String toSelectAll() { return "SELECT * FROM ModelSQLite"; }

    public String toSearch(String var){...}
```

ModelSQLite › toInsert()

ModelSQLite

```java
package com.rbd.roomvsrbdlite.model;

import androidx.room.ColumnInfo;
import androidx.room.Entity;

import java.io.Serializable;

@Entity(tableName = "ModelRoom")
public class ModelRoom implements Serializable{
    @ColumnInfo(name = "var1")
    String var1;
    @ColumnInfo(name = "var2")
    String var2;
    @ColumnInfo(name = "var3")
    String var3;
    @ColumnInfo(name = "var4")
    String var4;

    /* Generate Constructor, Getter dan Setter sama
       antara SQLite, Room, Maupun RBDLiTe */
}
```

ModelRoom

ModelRoom

```java
package com.rbd.roomvsrbdlite.model;

public class ModelRBDLiTe {
    String var1,var2,var3,var4;

    /* Generate Constructor, Getter dan Setter sama
       antara SQLite, Room, Maupun RBDLiTe */

    @Override
    public String toString() {...}

}
```

ModelRBDLiTe

Gambar 8. Perbandingan model pada SQLite, Room, dan RBDLiTe

c. RBDLiTe

RBDLiTe pada umumnya hanyalah alat bantu pembuatan basis data menggunakan SQLite. Dengan menggunakan RBDLiTe, pengguna hampir tidak perlu memasukan query sama sekali seperti pada Gambar 8 bagian "ModelRBDLiTe". Jumlah jangkauan fungsinya sangat sedikit hanya bisa digunakan untuk program yang sederhana. Sangat cocok untuk pemula karena membuat program android akan terasa sangat mudah dengan bantuan RBDLiTe. Untuk program yang lebih kompleks RBDLiTe bisa dipadukan dengan Room maupun SQLite sehingga tidak perlu mempelajari ulang semuanya. Pelajari RBDLiTe untuk membuat program sederhana dan jika ingin dikembangkan pelajarilah Room maupun SQLite.

4. KESIMPULAN

Untuk mempelajari penggunaan basis data internal pertama kali, disarankan menggunakan RBDLiTe karena penggunaannya sangat mudah sehingga dapat meningkatkan minat untuk mempelajari pembuatan aplikasi android. Kekurangan RBDLiTe dibandingkan dengan Room dan SQLite adalah instalasi yang cukup rumit dan jangkauan fungsi yang sangat sedikit, sehingga untuk membuat aplikasi yang kompleks, dianjurkan untuk menggabungkannya dengan SQLite maupun Room yang memiliki jangkauan fungsi yang lebih luas namun cukup sulit untuk dipelajari,

REFERENCES

[1] S. Lee, "Creating and using databases for Android applications," *Int. J. Database theory Appl.*, vol. 5, no. 2, pp. 99–106, 2012.
[2] N. Smyth, *Android Studio 3.6 Development Essentials - Java Edition: Developing Android 10 (Q) Apps Using Android Studio 3.6, java and Android Jetpack*. Payload Media, 2020.
[3] S. Al-Imamy, J. Alizadeh, and M. A. Nour, "On the Development of a Programming Teaching," *J. Inf. Technol. Educ. Innov. Pract.*, vol. 5, pp. 271–283, 2006.
[4] T. Yuen and M. Liu, "A cognitive model of how interactive multimedia,authoring facilitates conceptual understanding of object-Oriented programming in novices," *J. Interact. Learn. Res.*, vol. 22, no. 3, pp. 329–356, 2011.
[5] P. K. Schank, M. C. Linn, and M. J. Clancy, "Supporting Pascal programming with an on-line template library and case studies," *Int. J. Man. Mach. Stud.*, vol. 38, no. 6, pp. 1031–1048, 1993, doi: 10.1006/imms.1993.1048.
[6] D. Akopian, A. Melkonyan, S. C. Golgani, T. T. Yuen, and C. Saygin, "A Template-Based Short Course Concept on Android Application Development," *J. Inf. Technol. Educ. Innov. Pract.*, vol. 12, pp. 013–028, 2013, doi: 10.28945/1764.