Designing Application For Defect Recording and Handover Of Property Based On Mobile Application by Applying SQLite Technology

Persis Haryo Winasis*[1], Raga Maulana[2], Yodi Susanto[3]
[1,2,3]Magister of Computer Science Budi Luhur University
Email: [*1]**persis.winasis@gmail.com**, [2]ragamaulana1@gmail.com, [3]yodi.susanto@gmail.com

Abstract

Property development companies that produce housing products, high rise dwellings, and office buildings generally have data on the quality of buildings, one of which is obtained during the defect inspection process between developers and consumers before handing over units. Recording data is generally still done manually using a form on a paper. For these conditions, researchers tried to build an application based on mobile apps to digitally record the defect checklist of the dwelling so that the data collected can be processed for the needs of analysis and development strategies. Difficulties encountered during the unit handover process using digital methods on the newly completed property, one of which is the quality of data and internet signals. Mobile apps certainly require a data signal connection to send data to the server. This Android-based mobile apps will implement SQLite technology which allows the recording of transactions to be done locally first, which can then be synchronized into the database server after getting the required internet data connection. SQLite was chosen because it has a relatively small library code unlike relational DBMS in general. SQLite is also easy to use without complex configurations. With the support of the ease of function of SQLite it also allows applications to be integrated with the property sales application system.

***Keyword** — Property, Mobile apps, Android, Defect Checklist, SQLite*

1. INTRODUCTION

The property business in Indonesia is currently growing very rapidly, especially in big cities. This is due to the high number of unmet housing needs. For this reason, many property developers build vertical housing as a solution to meet these needs and can also optimize the need for green open space. In one building, an apartment will have a large number of units. With the large number of units, it becomes an obstacle in the process of examining defects on units and in the process of handing over units to consumers. It takes a large amount of paper and handwritten notes which may not be read properly and have no standardization in the handover process.

The limitation of internet data signal connection in an apartment that is not fully occupied by residents is also an obstacle in implementing a system that stores data directly to the server. Data signal providers tend not to invest in installing data signal amplifiers / transmitters in buildings where active occupancy rates tend to be low. On the basis of these problems, the researchers tried to make a design application for the defect list handover and the android-based unit handover process that can be used locally using SQLite which can be used to record and store data during the unit handover process. SQLite was chosen because apart from being lightweight, SQLite was deemed capable of accommodating user needs to record transaction data without having to connect directly to the database server due to the limited connectivity of data signals in new residential areas.

As a relational database management system that is ACID - compliant and has a relatively small code library size, SQLite is able to implement almost all standard elements that apply to SQL such as atomic transactions, database consistency, isolation, and durability (ACID), triggers, and also run complex queries.

2. RESEARCH METHODS

2.1 Previous Research

Referring to previous research that has been done, the use of SQLite has proven to be an alternative in handling data with a lighter process,easy to use, supported by multiplatform and flexibility of dynamic query capabilities. The research that is used as a reference can be seen in the following Table 1.

Table 1. Previous Research

Author (years)	Title	*Platform*
Angga Setiyadi, Tati Harihayati. (2015)	SQLite Application in the Examination and Presentation Timing Application	Web Application
Harni Kusniyati, Nicky Saputra Pangondian Sitanggang. (2016)	Android-Based Toba Samosir Culture Education Application	Android
Zeni Muhamad Noer, Kafi ngamali. (2017)	Application of the Library of SMK Siliwangi AMS Banjarsari based on Android	Andorid
Galeh Prehandayana, Widhi Yahya, Heru Nurwarsito. (2018)	Implementation of Data Dictionary Structures for Monitoring Systems for Internet Of Things Devices KRIPTOGRAFI ELGAMAL	Web Application

2.2 Method of Collecting Data

Researchers gather information by collecting data sourced from literature, journals and other sources related to the concept of SQLite and Android-based mobile apps. In addition, the researcher also collects existing defect recording data and property unit handover documents obtained from property development companies and conducts interviews with related users regarding the implementation of the handover of property units that are already running using manual methods.

2.3 System Development Method

To develop this system and application, researchers used the Waterfall method as shown in figure 1.

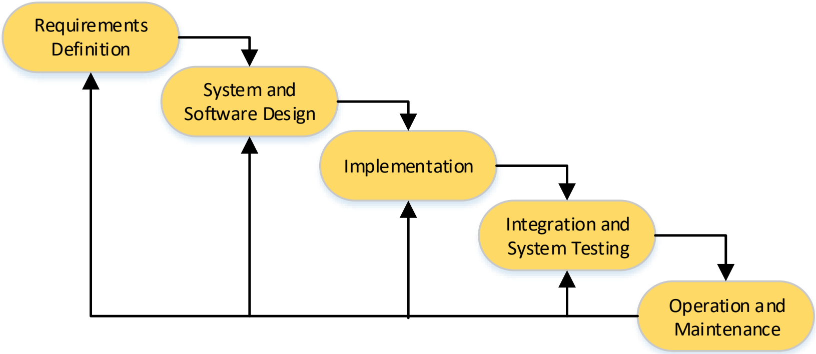

Requirements Definition
System and Software Design
Implementation
Integration and System Testing
Operation and Maintenance

Figure 1. Waterfall Method

3. RESULTS AND DISCUSSION

3.1 Requirements Definition

The analysis is carried out on the needs of the entire system to be built, because the software application will interact with various other elements such as data network connections, databases and hardware. At this stage, as much information as possible is collected from the user's needs, both through documentation and interviews.

3.2 System and Software Design

This stage performs the design and modeling of data structures, interfaces, software architecture and program algorithms. The results of this stage are in the form of a document called a software requirement, which will be used by programmers in making software applications.

3.2.1. Activity Diagram

The activity diagram is an overview of the functionality flow of a system. This activity diagram will illustrate various activity streams aimed at finding out the process flow in the defect recording application and the handover process of unit properties made. Details of the activity diagram for this application can be seen in Figure 2 to Figure 6 below.

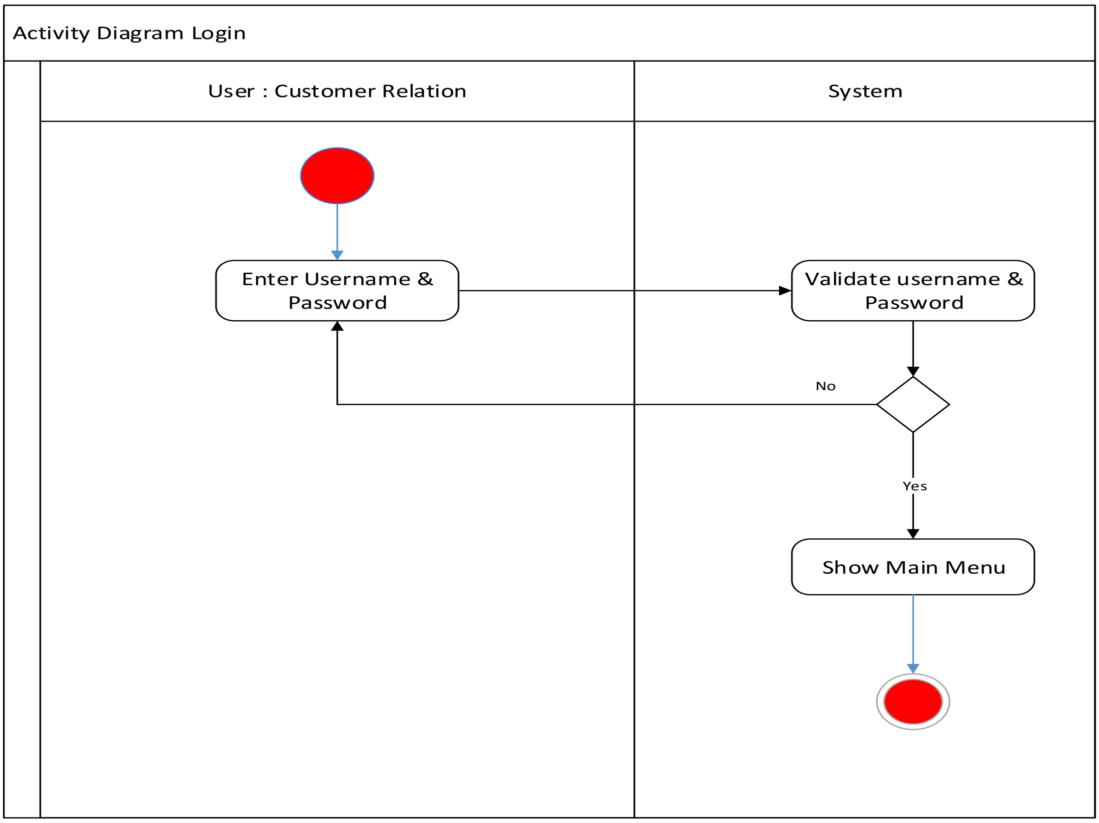


Figure 2. Activity Diagram Login Menu

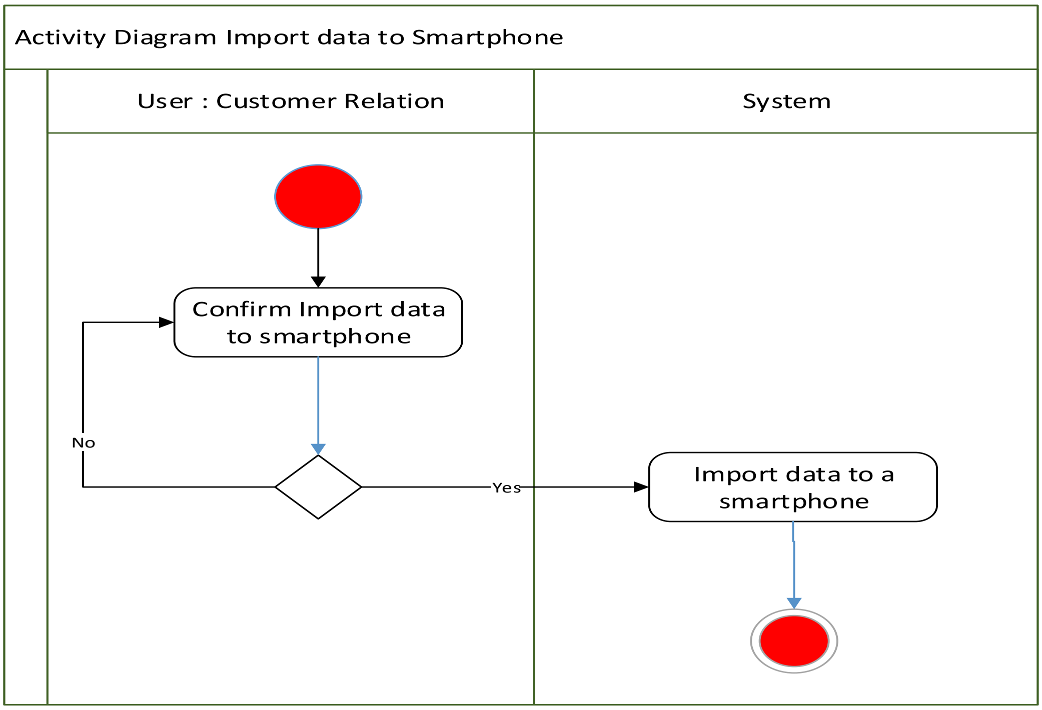

Activity Diagram Import data to Smartphone
User : Customer Relation
System
Confirm Import data to smartphone
No
Yes
Import data to a smartphone

Figure 3. Activity Diagram Import Data to Smartphone

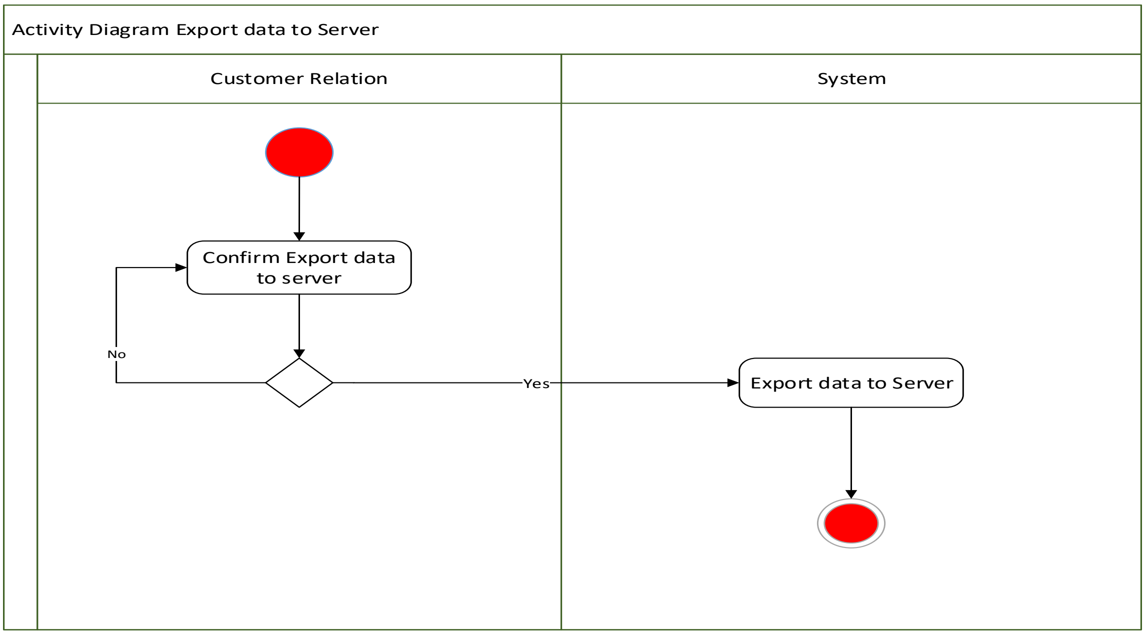

Activity Diagram Export data to Server
Customer Relation
System
Confirm Export data to server
No
Yes
Export data to Server

Figure 4. Activity Diagram Export Data to Server

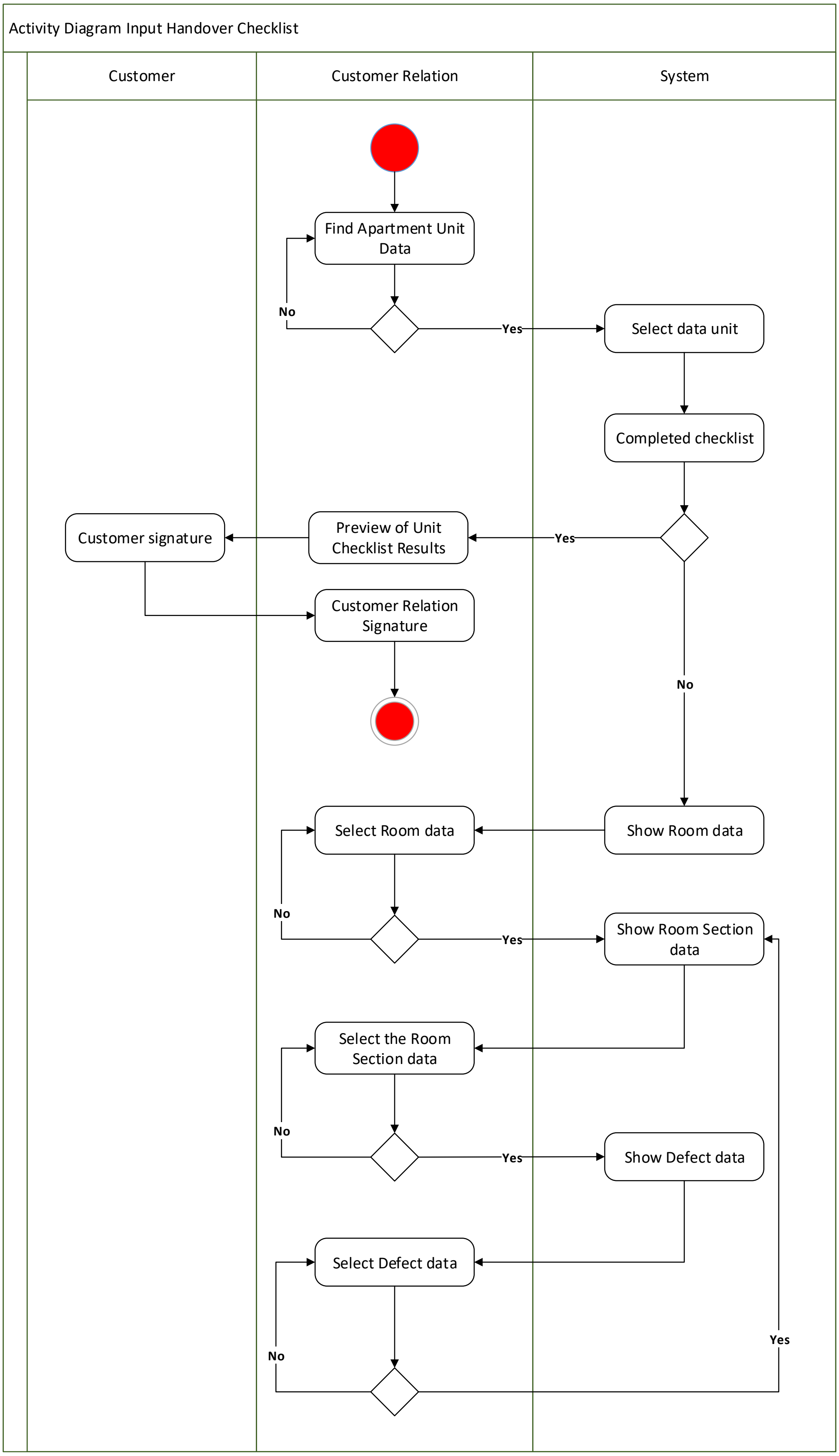


Figure 5. Activity Diagram Activity Input The Handover Checklist

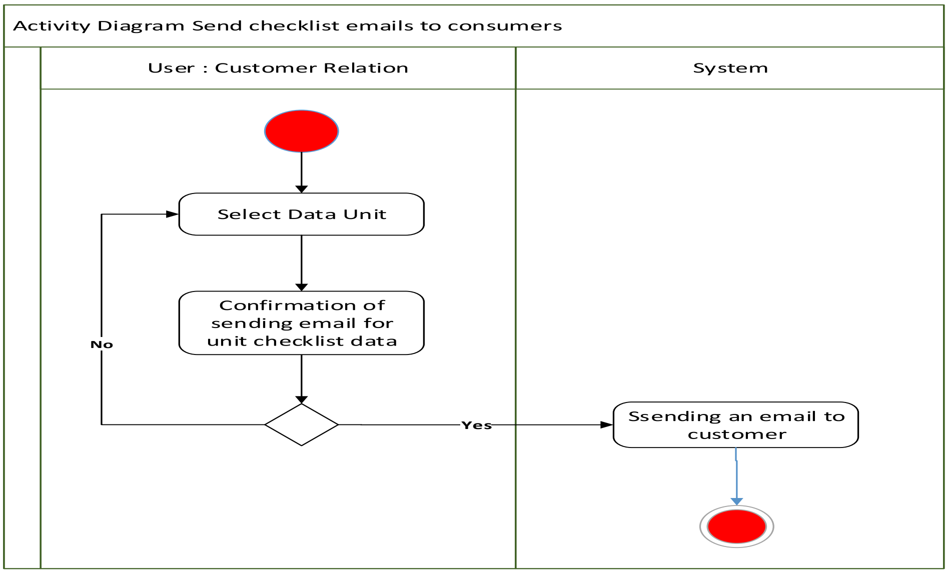

Activity Diagram Send checklist emails to consumers
User : Customer Relation
System
Select Data Unit
Confirmation of sending email for unit checklist data
No
Yes
Ssending an email to customer

Figure 6. Activity Diagram Send Email to Customer

3.2.2. Class Diagram

Class diagrams describe the structure of the system in terms of defining classes that are made to build a system. The classes in the system structure must be able to perform functions according to system requirements. In the following Figure 7, the definitions of classes in building the defect recording application and the property unit handover process will be displayed.

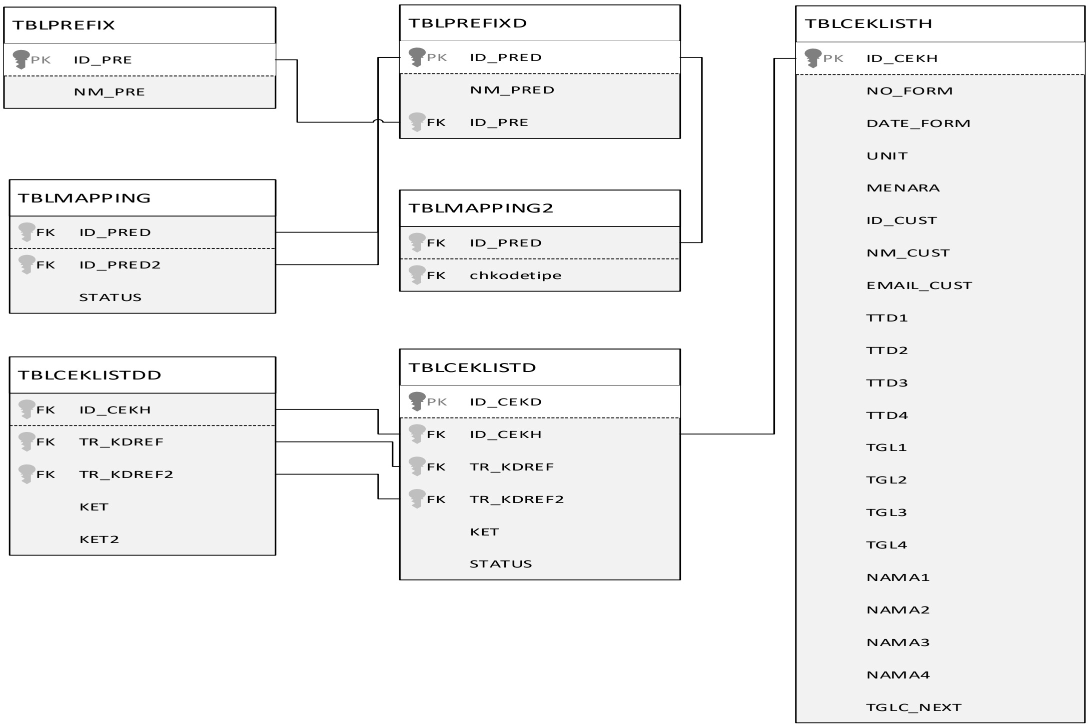
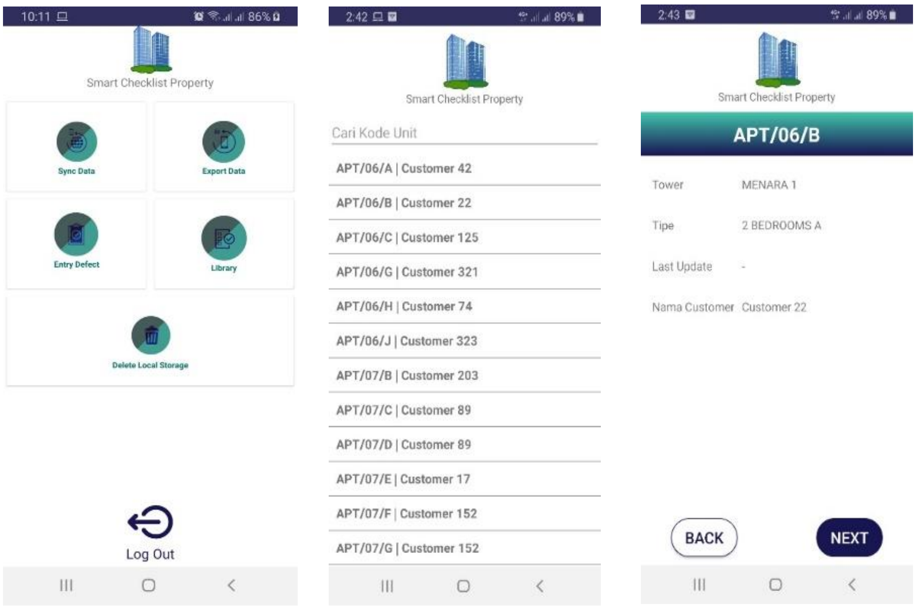


Figure 7. Class Diagram

3.2.3. User Interface

The Graphical User Interface (GUI) that was built has gone through a needs analysis process and is easy to use. Here are some GUI views for defect recording applications and property unit handover. The initial stage is the selection of property units to be recorded, details of property units will appear as validation media. The main menu can be seen in Figure 8 below.



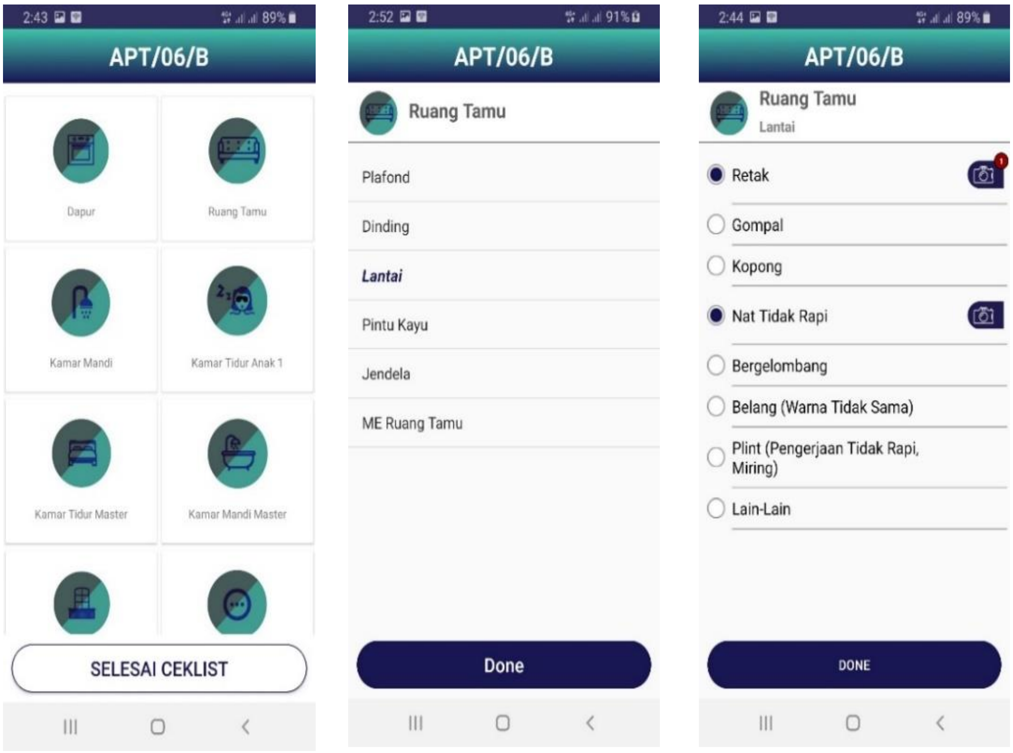
APT/06/A | Customer 42
APT/06/B | Customer 22
APT/06/C | Customer 125
APT/06/G | Customer 321
APT/06/H | Customer 74
APT/06/J | Customer 323
APT/07/B | Customer 203
APT/07/C | Customer 89
APT/07/D | Customer 89
APT/07/E | Customer 17
APT/07/F | Customer 152
APT/07/G | Customer 152

APT/06/B

Tower	MENARA 1
Tipe	2 BEDROOMS A
Last Update	-
Nama Customer	Customer 22

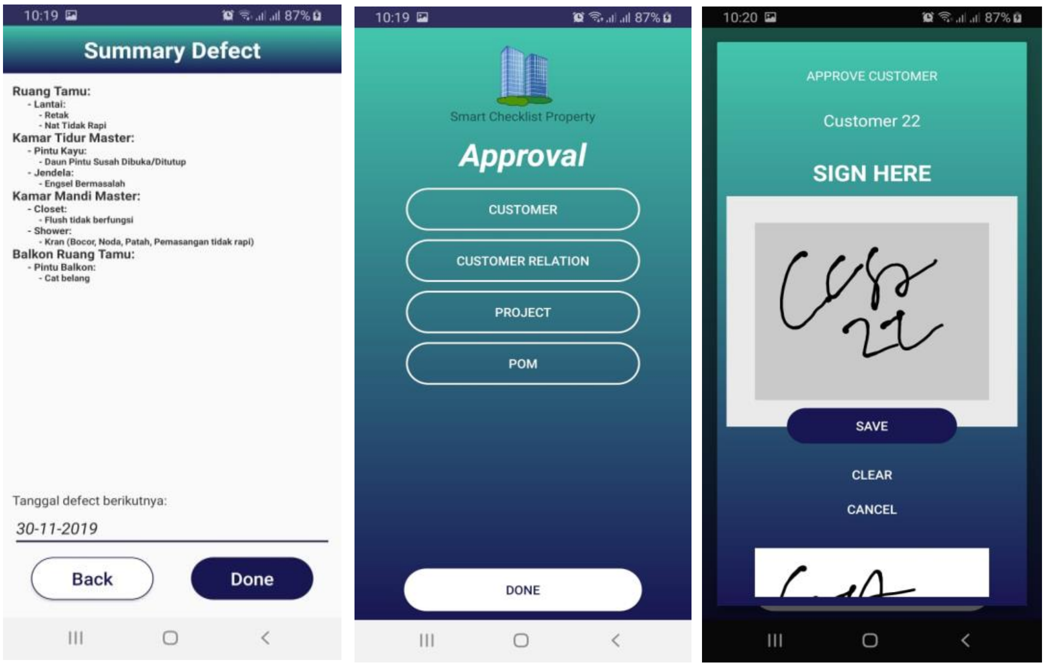
Figure 8. Graphical User Interface (1)

The next stage is the selection of the area of the room which is then followed by the selection of items that will be recorded for the defect. Categories and types of defects have been made templates according to the defects that often appear in a property. This aims to facilitate the classification at the analysis and reporting stage. The recording of the defect will be accompanied by a photo upload process related to the damaged item to facilitate repair and can be used as a comparison at the next inspection with the same item, which then the parties will agree to determine the next defect inspection schedule for inspection and re-recording of defects on the property unit has been repaired. This recording process is supported by a tiered approval system by related parties such as consumers, customer relations, the project team and property managers. After all processes are fulfilled, the application will send an email to the related party as proof of the report. The intended menu can be seen in Figures 9 and 10 below.



● Retak
○ Gompal
○ Kopong
● Nat Tidak Rapi
○ Bergelombang
○ Belang (Warna Tidak Sama)
○ Plint (Pengerjaan Tidak Rapi, Miring)
○ Lain-Lain

Figure 9. Graphical User Interface (2)



Ruang Tamu:
- Lantai:
  - Retak
  - Nat Tidak Rapi

Kamar Tidur Master:
- Pintu Kayu:
  - Daun Pintu Susah Dibuka/Ditutup
- Jendela:
  - Engsel Bermasalah

Kamar Mandi Master:
- Closet:
  - Flush tidak berfungsi
- Shower:
  - Kran (Bocor, Noda, Patah, Pemasangan tidak rapi)

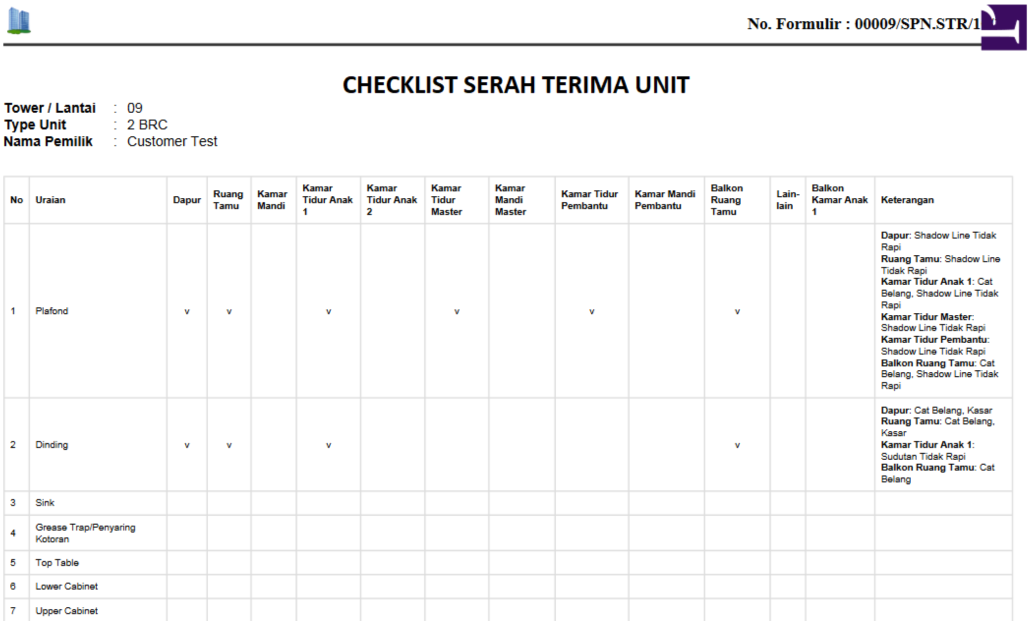
Balkon Ruang Tamu:
- Pintu Balkon:
  - Cat belang

Tanggal defect berikutnya: 30-11-2019

Customer 22

Figure 10. Graphical User Interface (3)

After making sure that the property unit is no longer defective, the developer can process the unit handover to consumers with the same application. This process is also documented by the application in the form of a photo, digital signature and email. As a result, all records starting from the initial recording process to the status where the unit is declared no longer have defects can be processed by the property developer in the form of a report as shown in Figure 11 below which can later be reprocessed for development analysis needs.



No. Formulir : 00009/SPN.STR/1

CHECKLIST SERAH TERIMA UNIT

Tower / Lantai : 09
Type Unit : 2 BRC
Nama Pemilik : Customer Test

No	Uraian	Dapur	Ruang Tamu	Kamar Mandi	Kamar Tidur Anak 1	Kamar Tidur Anak 2	Kamar Tidur Master	Kamar Mandi Master	Kamar Tidur Pembantu	Kamar Mandi Pembantu	Balkon Ruang Tamu	Lain-lain	Balkon Kamar Anak 1	Keterangan
1	Plafond	v	v		v		v		v		v			**Dapur**: Shadow Line Tidak Rapi **Ruang Tamu**: Shadow Line Tidak Rapi **Kamar Tidur Anak 1**: Cat Belang, Shadow Line Tidak Rapi **Kamar Tidur Master**: Shadow Line Tidak Rapi **Kamar Tidur Pembantu**: Shadow Line Tidak Rapi **Balkon Ruang Tamu**: Cat Belang, Shadow Line Tidak Rapi
2	Dinding	v	v		v						v			**Dapur**: Cat Belang, Kasar **Ruang Tamu**: Cat Belang, Kasar **Kamar Tidur Anak 1**: Sudutan Tidak Rapi **Balkon Ruang Tamu**: Cat Belang
3	Sink													
4	Grease Trap/Penyaring Kotoran													
5	Top Table													
6	Lower Cabinet													
7	Upper Cabinet													

Figure 11. Reporting

3.3 *Implementation*

This stage is the creation of an application using the programming code specified in the software requirements document. At this stage, the creation of a database and infrastructure that supports the system to work well is also carried out. The result at this stage is a beta version of the software. The following is the implementation of the SQLite programming code using the Java programming language and the Android Studio IDE in this study:

- Create *Class DatabaseHelper*
  At this stage, create a java class named DatabaseHelper with extends SQLiteOpenHelper, where the DatabaseHelper class acts as a liaison when executing CRUD commands (CREATE, UPDATE, DELETE) and SELECT on the Activity or Fragment.
- Execute Commmand *CREATE*
  Use function .execSQL(String sql);
  Example:
  ```
  String sql = “CREATE TABLE table1 (field1 INT, field2 CHAR(15))”;
  objDatabaseHelper.execSQL(sql);
  ```
- Execute Commmand *SELECT*
  Use function .rawQuery(String sql, String[] selelctionArgs) yang ditampung dalam *class* Cursor.
  Example:
  ```
  Cursor cursor = objDatabaseHelper.rawQuery(“SELECT * FROM table1”);
  ```
- Execute Commmand *INSERT*
  Use function .insert(String table, String nullColumnHack, ContentValues values);
  Example:
  ```
  ContentValues values = new ContentValues();
  values.put("Field1", "001");
  values.put("Field2", "Budi");
  values.put("Field3", "Luhur");
  objDatabaseHelper.insert(NAMA_TABLE, null, values);
  ```
- Execute Commmand *UPDATE*
  Use function update(String table, ContentValues values, String whereClause, String[] whereArgs);
  Example:
  ```
  ContentValues cv = new ContentValues();
  cv.put("Field1","Budi");
  cv.put("Field2","Luhur");
  objDatabaseHelper.update(NAMA_TABLE, cv, "_id = ?", new String[]{var_id});
  ```

- Execute Commmand *DELETE*
  Use function delete(String table, String whereClause, String[] whereArgs);
  Example:
  objDatabaseHelper.delete(NAMA_TABLE,"id=? and name=?",new
  String[]{"001","Budi"});

3.4 Integration and System Testing

All modules and functions in the software that have been built begin to be integrated and tested, in order to avoid errors and the results released from the application are able to achieve user needs.

3.5 Operation and Maintenance

Maintenance of software is required, such as developing features and modules according to future needs. And when running, it may still find gaps in application errors that have never been found before.

4. CONCLUSION

From the results of design and research that has been carried out on the application of recording defects and handover of property units, it can be concluded that the application of SQLite in this application has succeeded in making it easier to record defects and handover units by utilizing local data storage which is then synchronized into the database. server. This function is very useful for areas that do not have an internet connection in sending data to the database. In addition, this application can speed up the process of recording defects and handing over units and has a standardization in the recording process, compared to manual recording using paper. The resulting data can be processed by the developer to review the construction of a building in combination with the project data, including contractors, length of construction, costs incurred, and others. The processed data can then be used by property developers to determine development strategies for further building projects, so as to optimize costs, time and contractor selection in the process.

5. SUGGESTED

Suggestions for developing a defect checklist recording system and handover of apartment units include the creation of a system module that will be used when the construction phase begins. The participation of the contractor and the developer in conducting defect checks during the construction phase will reduce the risk of many defects at the time of handing over the unit to consumers because the inspection has been carried out intensively when construction begins. Another thing is, it can integrate this system with a development monitoring and budgeting system, so that it will obtain data analysis of the incurred construction costs, the length of time of development and the development history.

REFERENCES

[1] KBBI, “Kamus Besar Bahasa Indonesia (KBBI),” Kementerian Pendidikan dan Budaya. 2016.

[2] “Android Programming,” Netw. Secur., 2015.J. Andi, “Development of a Child Tracker Application Based on Assisted - Global Positioning System (A-GPS) with the Android Platform,” J. Ilm. Komput. dan Inform., vol. 1, no. 1, pp. 1–8, 2015.

[3] S. T. Bhosale, M. T. Patil, and M. P. Patil, “SQLite: Light Database System,” Int. J. Comput. Sci. Mob. Comput., 2015.

[4] A. SETIYAD and T. Harihayati, “Application Of Sqlite In The Exam And Presentation Timing Application,” Maj. Ilm. UNIKOM, 2015.

[5] H. Kusniyati and N. S. Pangondian Sitanggang, “Android Based Toba Samosir Culture Education Application,” J. Tek. Inform., 2016.

[6] A. Sukmaindrayana and R. Sidik, “Siliwangi Ams Banjarsari Public School Library Application Based on Android,” J. Manaj. KINERJA ISSN 2407-7305, vol. 4, no. 2, pp. 41–50, 2017.

[7] G. Prehandayana, W. Yahya, and H. Nurwasito, “Implementation of Data Dictionary Structure for Internet of Things Device Monitoring System,” J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya, vol. 2, no. 10, pp. 3466–3473, 2018.