Implementation of Realtime Database for IoT Home Automation and Energy Monitoring Apps based on Android

Enggar Swahadika
Informatic and Computer Engineering
Politeknik Elektronika
Negeri Surabaya
Surabaya Indonesia
enggarswahadika@gmail.com

Adnan Rachmat Anom Besari
Informatic and Computer Engineering
Politeknik Elektronika
Negeri Surabaya
Surabaya, Indonesia
anom@pens.ac.id

Iwan Kurnianto Wibowo
Informatic and Computer Engineering
Politeknik Elektronika
Negeri Surabaya
Surabaya Indonesia
eone@.pens.ac.id

***Abstract*—Nowadays the concept of IoT (Internet of Things) and its development isn't a new thing anymore for us. One of its part which should be considered to build an internet of things platform is how the data will be stored and distributed on the internet. Database become a necessary part because the operation data between IoT device and the interface such as mobile or web application occur and stored here. For its implementation an IoT platform should be developed with high accuracy and minimum execution time. This paper will discuss about implementation of real-time database for home automation and monitoring energy application and its system design involved the protocol data, how the data have been proceed and stored to the database. Furthermore this paper will show the performance of database which its designed.**

Keywords—Internet of Things, Real-time database, Home automation, Energy monitoring

I. Introduction

Nowadays, the concept of IoT (Internet of Things) and its development isn't a new thing anymore for us. Much research in recent year are focus on its development. With this technology, many devices can be connect to the internet. Because of it the data from each devices can be acquire and analyse furthermore these devices can be control anywhere. Implementation of Internet of Things can be apply on many sector such as smart city, smart public transportation system, digital payment, manufacture, retail, logistic, etc. [1]

Additionally, to build or develop Internet of Things platform we need to consider a few things. The part including the rule engine to manage action from device through sensor data and condition, database for storing the data, interface platform such as mobile or web apps which interact directly with user, SDK or library support as third party. [2] From those consideration listed above, the part will be necessary is database. Data from all devices process will be stored here so we need to design our database structure so that the minimum percentage data loss and faster execution time can be reached.

In this paper, we will implement real-time database architecture to build home automation and energy monitoring on mobile application based on Internet of Things. Furthermore, the other parts to build it will be included too such as design protocol data, real-time database will be use, retrieval and post data process and the integration with the mobile application itself. The result of implementation will be shown at the end of content.

II. Materials and Method

To develop the application there are several materials need to be prepared. The first we need to prepare is the IDE to create the application itself. The second one is real-time database, the real-time database which implement on this paper is Firebase database from Google because its flexibility and SDK support. Then the last is POJO (Plain Old Java Object) Model to handle retrieving, write, update and remove data operation on application side. Method that used to create rule engine is ECA (Event Condition Action) Rule, which provide the functionality of device listed on application.

A. Android Studio IDE

Android Studio is the official Integrated Development Environment (IDE) for Android application development based on IntelliJ IDEA. Android studio provide many feature such as flexible Gradle-based build system, Code template and GitHub integration, testing tool and frameworks, C++ and NDK support and the important is android studio is built-in support for Google Cloud Platform that ease us to integrate our application with cloud service such as database, engine-app or cloud messaging [3].

B. Firebase Cloud System

Firebase is a combination of Google's cloud platform services, including instant messaging, user authentication, real-time database, storage, hosting and so on. This paper mainly uses its real-time database service to build entire apps system. The main function of firebase real-time database service is its handle the data collection. Based on Firebase Cloud, controller were directly connected to its database to retrieve or write data. Firebase provides access to its database in multi-platform devices such as IOS, Android, Java Script, and REST API [4]. For this paper we will use Android SDK to integrate our mobile apps with firebase real-time database.

C. POJO (Plain Old Java Object)

Model Data Driven (MDD) in mobile development can simplify the development of mobile apps and reducing significantly technical complexity. MDD applied Model, View and Controller (MVC) pattern of mobile application. The main point of this pattern is the model transformation of

978-1-7281-4449-8/19/$31.00 ©2019 IEEE

POJO class for model layer, which handle the data operation of the application. For its implementation, POJO model related to one model data from database such as data type on each database attribute, nodes structure and it allocation itself [5].

D. *ECA (Event Condition Action) Rule*

ECA (Event Condition Action) Rule is the form **on** *event,* **if** *condition* **do** *action* [6]. Simple ECA rule can be used for non-technical users and ease them to configure and control home automation system. Furthermore ECA Rule can be used to describe which action performed from devices in response to event and condition hold [7]. Rule design and its implementation will shown on the system design section.

III. System Design

This section discusses about the system design of this research .

A. *System Diagram*

Home automation and energy monitoring application has a purpose to controlling device and monitoring its energy usage. For implement its function, we need to create data model for each function. The data model included the process will be executed for each function, so that the data can be stored on the database through POJO Model. System diagram of this system will shown on fig 1.

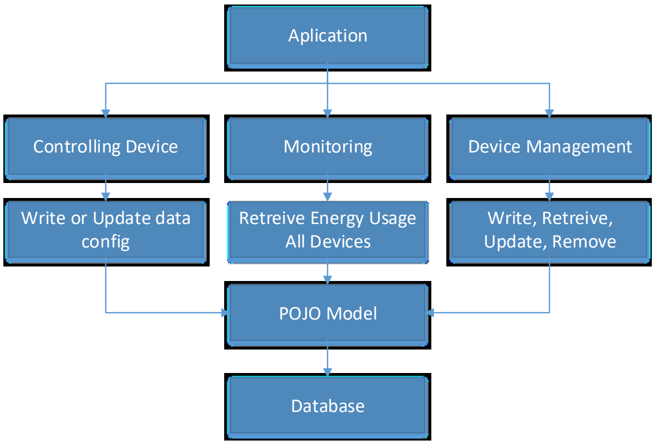

Aplication
Controlling Device
Monitoring
Device Management
Write or Update data config
Retreive Energy Usage All Devices
Write, Retreive, Update, Remove
POJO Model
Database

Fig. 1. System diagram

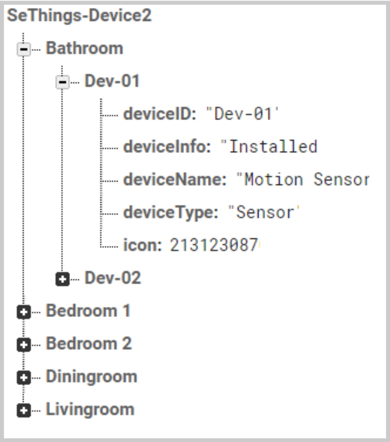
The POJO Model will allocate the data into database in accordance with the functions, which own by the application. Monitoring process will execute retrieve data usage all devices from database, and then device management process will execute (retrieve, write, update and remove) device data from database, last controlling device process will execute (write and update) data configuration device from database.

B. *Database Structure*

Firebase was built using *no SQL* database, which design to allocate data on a specific branch (node) and has no relationship between tables as in SQL database. The database created on this research has seven nodes, which are used for data communication purposes. Explanation below will show the structure of database used .

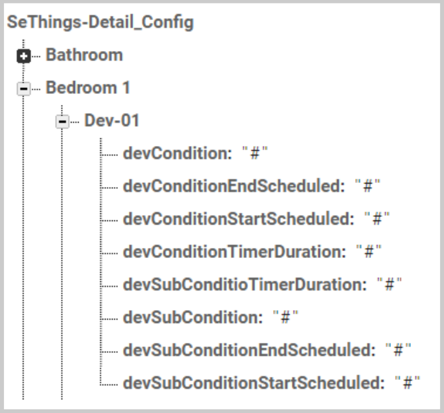
- Device data is the node on database which its function to store device data installed. Attribute or children of this node such as device name, type, info are placed on room name with device id as key to access the data
- Device usage is node on database which its function to store device energy usage. This node designed for purpose display energy usage from all devices, energy usage from each room or each device
- Device daily usage is the nodes in a database used to store daily data on the use of each device. This node is designed to do the estimate the average of energy usage by calculating total usage and perform in accordance with the amount of child (node) on each date
- Device configuration is the nodes on the database which its used to store configuration data device appropriate input from the application. Each node stored information about how the device are configured. Information contain the device id of the connected sensor and the type of action performed on the device that is the action that is performed once or repeated process. This node will be directly associated another nodes the nodes is device configuration detail to store the information about details configuration of a device.
- Device detail configuration is the nodes on the database that is directly related with device configuration node. This node store the details condition of configured device
- Sensor value is the nodes on the database which its used to store values of the sensor. this node will store values from sensors that continually update
- User list is the nodes on the database which its used to store user data and designed to give access for user in purpose of use the application.

The example of database structure will be shown on fig 2 and fig 3 [8].



```
SeThings-Device2
  Bathroom
    Dev-01
      deviceID: "Dev-01'
      deviceInfo: "Installed
      deviceName: "Motion Sensor
      deviceType: "Sensor'
      icon: 213123087
    Dev-02
  Bedroom 1
  Bedroom 2
  Diningroom
  Livingroom
```

Fig. 2. Database structure on device node



```
SeThings-Detail_Config
  Bathroom
  Bedroom 1
    Dev-01
      devCondition: "#"
      devConditionEndScheduled: "#"
      devConditionStartScheduled: "#"
      devConditionTimerDuration: "#"
      devSubConditioTimerDuration: "#"
      devSubCondition: "#"
      devSubConditionEndScheduled: "#"
      devSubConditionStartScheduled: "#"
```

Fig. 3. Database structure on device node

C. *Rule Design*

The Protocol is a rule which defines several functions that exist in a network that must be met by the side of the sender and the receiver in order to make the communication process can take place properly. The Protocol is generally used on a real-time communication standard which is used to set the structure of information for long term storage on database.

The rule are designed based on ECA(Event Condition Action) Rule which is separate device configuration into 3 part. On this research there are several rule are designed to configure a device, there are direct on/off manually, timer, schedule, and sensor integration. The explanation below will show the design rule of each configuration.

- Switch on / off
- Timer configuration rule
- Schedule configuration rule
- Sensor condition rule

Switch on configuration are the simplest one among the configuration listed above. The rule formed from this simple analogy, **on** *device id* target **if** *switch off* **then do** *switch on.* When its form into table will become

TABLE I. SWITCH ON / OFF CONFIGURATION RULE DESIGN

Rule	**Designed rule**		
	Event	***Condition***	***Action***
Switch on	Device-id	SWITCH_OFF	SWITCH_ON
Switch off	Device-id	SWITCH_ON	SWITCH_OFF

Then for timer configuration device there is additional duration parameter for its condition. The rule of timer configuration will shown at table 2.

TABLE II. TIMER CONFIGURATION RULE DESIGN

Rule	**Designed rule**		
	Event	***Condition***	***Action***
Timer	Device-id	TIMER#Duration	ON#While#Timer

Then for scheduled configuration device, there are two additional parameter. The first parameter is when device start to be switch on, and then the second parameter is when the device will be switch off. The rule of scheduled configuration will shown at table 3.

TABLE III. SCHEDULE CONFIGURATION RULE DESIGN

Rule	**Designed rule**		
	Event	***Condition***	***Action***
Schedule	Device-id	SCHEDULED	
		# START_TIME	SWITCH_ON
		# END_TIME	SWITCH_OFF

Last for sensor condition there are several parameter depend on type sensor related with device and action that will be executed. The rule of its configuration will shown at table 4.

TABLE IV. SENSOR CONFIGURATION RULE DESIGN

Rule	**Designed rule**		
	Event	***Condition***	***Action***
Sensor Integration	Device-id	SENSOR	TIMER Configuration
		# TYPE	SCHEDULED Configuration
		# VALUE	

from sensor integration configuration the action from device will inherit from timer and scheduled configuration. All of rule designed according table 1 until table for will be form into one string block state and send to the database on detail configuration node and its allocated automatically to each attribute according the configuration applied on the selected device.

D. *Data Operation Flow Process*

All nodes from database, which have its function and specification in order to operate properly we need a pre-processor language on the application. Because android application using java as pre-processor language, process of retrieving, write, update and delete data on the database we will need POJO model to handle it. The flow of POJO model usage will be shown on figure 4 until figure 7.

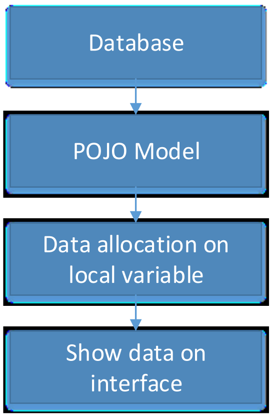


Fig. 4. Retrieve flow process from database

Identify applicable funding agency here. If none, delete this text box.

The process according figure 4 is retrieve process or get data from database. POJO model will get the raw data from database. After get raw data the POJO will allocated the data on local variable according its data type for each data retrieved.

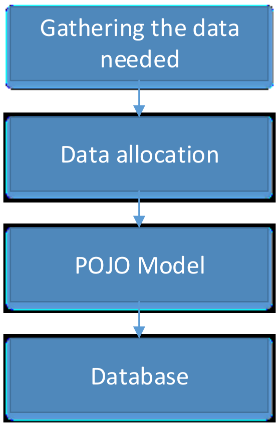


Fig. 5. Write data flow process

According figure 5, process write data beginning from gathering the data needed in specify nodes on the database. After gathering the data needed next process is allocating the data according each data types. After that POJO model will handle the rest to push the data on database.

Update data is similar to write data but on update data process, we need to specify the reference key of the updated data. Figure 6 will show the flow process of updating data.

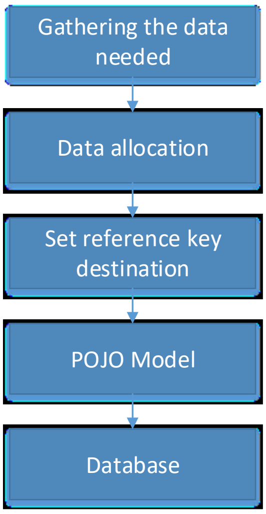
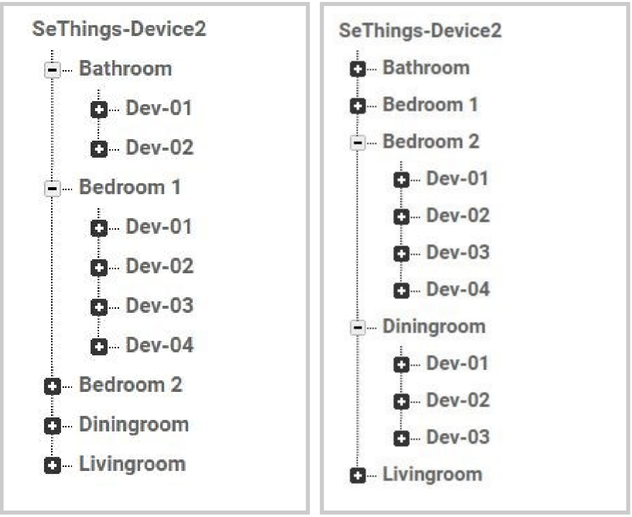


Fig. 6. Update data flow process

IV. Implementation Result

This section will show the result of real-time database implementation on home automation and energy monitoring application. For testing purpose, the data from database will be compared with the data shown on application interface. The process will be executed on each menu that involved data operation from database. Furthermore, for database reliable testing, the application will send configuration data for each device with different condition and action after that the percentage of data loss and the execution time for each process will be shown as graph for testing result.

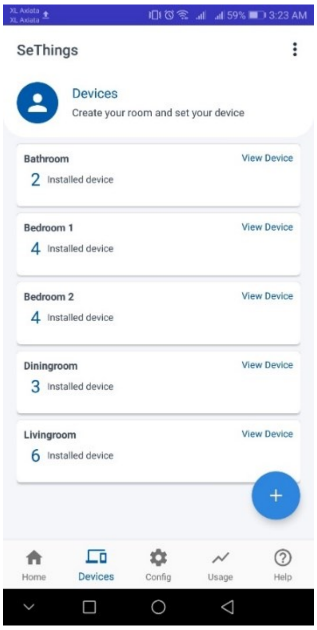
The first experiment is retrieve data device from database. Structure and the detail data will be shown on fig 7.



(A) (B)

Fig. 7. (A) Database Structure on device node, (B) Database Structure on device node

Fig 7 is explain about the data device which stored on device node. According fig 7 the total of child(data node separator), there are 6 room with different amount of installed device. According the process of retrieving data on fig 4 the result shown on the application interface should be covered all information about room and the installed device. The result on the application interface will be shown on fig 8.



SeThings

Devices
Create your room and set your device

Bathroom View Device
2 Installed device

Bedroom 1 View Device
4 Installed device

Bedroom 2 View Device
4 Installed device

Diningroom View Device
3 Installed device

Livingroom View Device
6 Installed device

Fig. 8. Application result

Based on data shown through application interface on fig 8 all data from database can be retrieve successfully. The number of installed device form each room are represent the children count from each node on database can be seen at fig 7.

The second experiment is test all device configuration process. Total of devices that driven by the application is 19 items according figure 8, but to test performance of each device the testing take the sample of lamp configuration. The testing process start from switch on/off and timer configuration until scheduled.

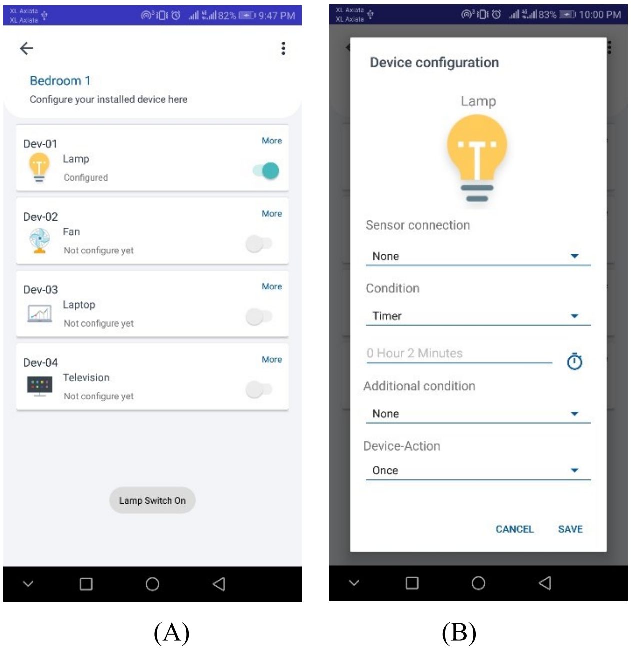


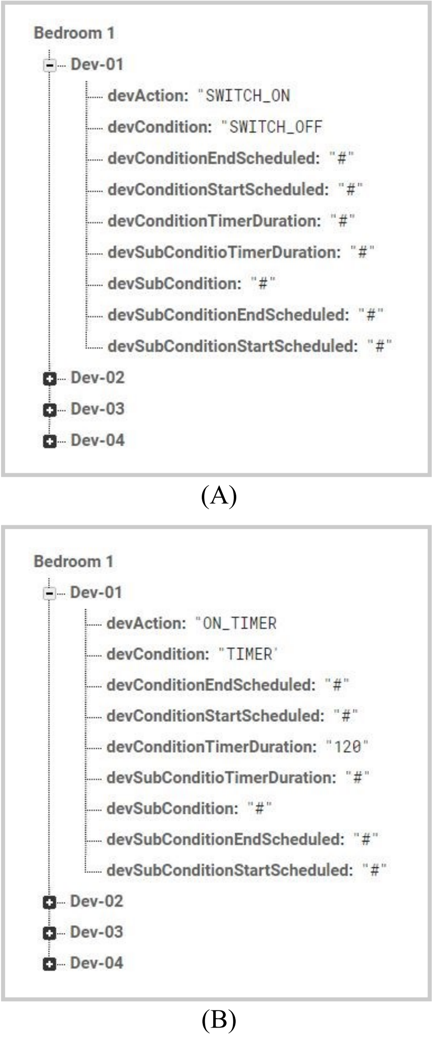
Fig. 9. (A) Apps interface switch on/off configuration, (B) Apps interface timer configuration

According design rule of device configuration, to write data configuration on database we must form the data according selected configuration like table 1 until table 4. From fig 9 , the selected configuration are switch on/off and timer configuration so the data that write on database must be consist on table 5.

TABLE V. CONFIGURATION DEVICE DATA ACCORDING APPS CONFIG

Rule	**Designed rule**		
	Event	***Condition***	***Action***
Switch on	Dev-01	SWITCH_OFF	SWITCH_ON
Timer	Dev-01	TIMER#120	ON#While#Timer

On the database the data will allocated into specific nodes to separate condition of configured device. The result of configuration will be shown on fig 10.



```
Bedroom 1
  Dev-01
    devAction: "SWITCH_ON
    devCondition: "SWITCH_OFF
    devConditionEndScheduled: "#"
    devConditionStartScheduled: "#"
    devConditionTimerDuration: "#"
    devSubConditioTimerDuration: "#"
    devSubCondition: "#"
    devSubConditionEndScheduled: "#"
    devSubConditionStartScheduled: "#"
  Dev-02
  Dev-03
  Dev-04
```

(A)

```
Bedroom 1
  Dev-01
    devAction: "ON_TIMER
    devCondition: "TIMER'
    devConditionEndScheduled: "#"
    devConditionStartScheduled: "#"
    devConditionTimerDuration: "120"
    devSubConditioTimerDuration: "#"
    devSubCondition: "#"
    devSubConditionEndScheduled: "#"
    devSubConditionStartScheduled: "#"
  Dev-02
  Dev-03
  Dev-04
```

(B)

Fig. 10. (A) Switch on/off configuration database structure (B) Timer configuration database structure

From fig 10 can be seen that the configuration data from switch on/off and timer configuration are allocated to each node consist on table 5 correctly.

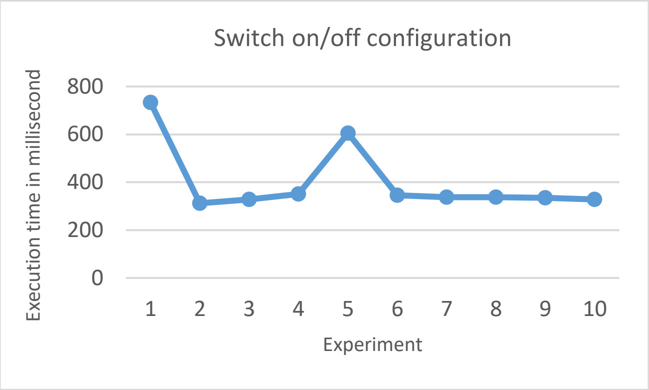
The data that write or retrieve on database to fulfil criteria, as real-time database there must be have a short execution time and have minimum data loss percentage. For its purpose the testing will be conducted by executing device configuration process to home simulator. The home simulator is application which it's made as installed device on database to read configuration data and do the action according configuration sent from application. Table 6 will show the percentage of successfully data can be executed by simulator device and its execution time.

TABLE VI. CONFIGURATION DEVICE DATA ACCORDING APPS CONFIG

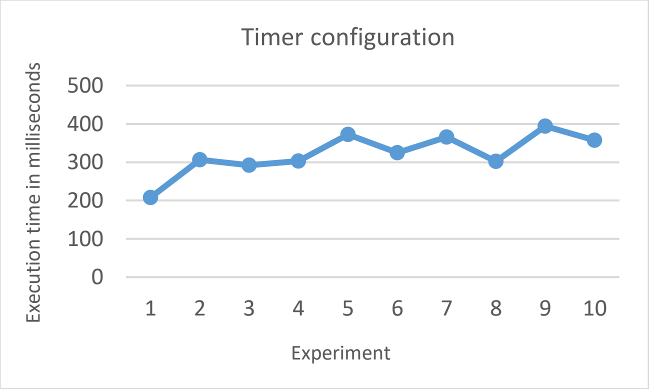
No	**Success rate experiment**				
	Success?	***Time I***	***Time II***	***Time III***	***Time IV***
1	✓	733 ms	208 ms	214 ms	125 ms
2	✓	312 ms	306 ms	195 ms	179 ms
3	✓	328 ms	292 ms	165 ms	172 ms
4	✓	351 ms	303 ms	164 ms	212 ms

No	Success rate experiment				
	Success?	***Time I***	***Time II***	***Time III***	***Time IV***
5	✓	605 ms	372 ms	227 ms	174 ms
6	✓	346 ms	325 ms	565 ms	512 ms
7	✓	337 ms	366 ms	474 ms	337 ms
8	✓	338 ms	302 ms	201 ms	401 ms
9	✓	335 ms	394 ms	213 ms	312 ms
10	✓	394 ms	357 ms	185 ms	254 ms

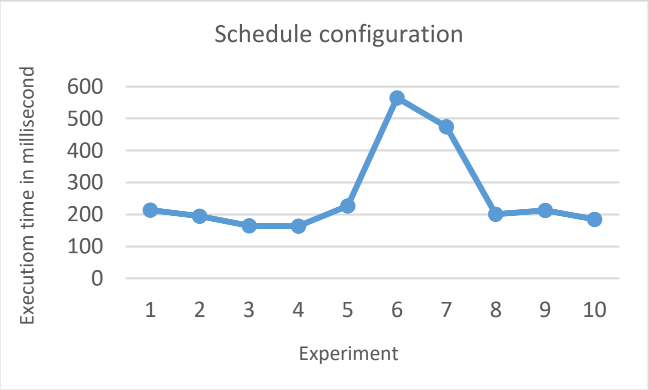
Timer I until IV are execution time form different configuration selected according rule design from table 1 until table 4. Figure 11 show the graphic of execution time from all testing process.



(A)



(B)

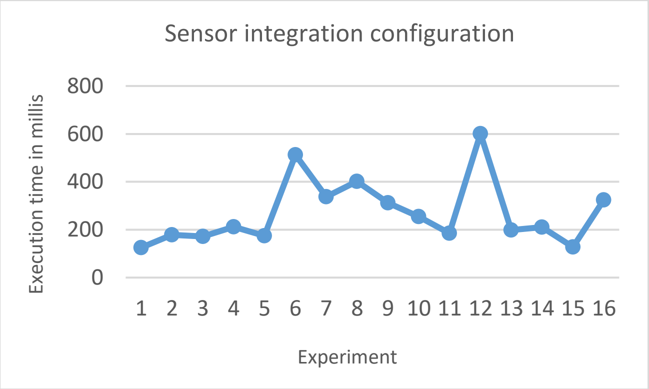


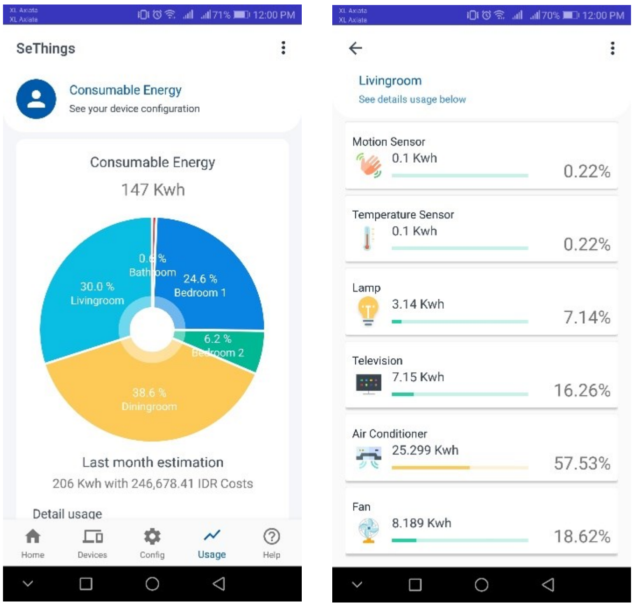
(C)
(D)

Fig. 11. (A) Execution time from switch on/off configuration, (B) Execution time from timer configuration, (C) Execution time from schedule configuration, (D) Execution time from sensor integration configuration

From data shown according figure 11 the complexity of data configuration send to the database aren't cause the execution time rise up. Real-time database can handle the operation data and keep the execution time to minimum value it can depend on the internet connection. From testing process according table 6, the total average of execution time from all configuration process is 316,52 milliseconds. The data loss of testing process are 0% based on all configuration process.

Last is result from monitoring process is show all energy usage from all installed devices in pie chart and details from each room according on fig 12. There are 2 interface to show the result of monitoring energy. The general monitoring energy is show the total energy consume from each room as graph with its percentage. The room selected will show the consumed energy from each device with its percentage too. The difference is bar notification to determine how much energy use from each device.





(A) (B)

Fig. 12. (A) General monitoring energy result (B) Sample from one room energy consume

V. Conclusion

To build Internet of Things platform real-time database can be use as the foundation of data storage system because its reliability to store data and give response as fast as data request held. Firebase is one of the real-time database service with many support of SDK in any platform from middle level system until application layer. Furthermore, the benefit of use this service can make your platform give data response base on request in real-time no matter how big data will proceed.

References

[1] Harbor Research, “What Exactly Is The Internet Of Things”, March 2014.

[2] Kiran Jot Singh and Divneet Singh Kapoor, ”Create Your Own Internet of Things”,15 March 2017.

[3] Android Studio overview, https://developer.android.com/studio/intro online reference

[4] Wu-Jeng Li, Chiaming Yen and You-Sheng Lin, “JustIouT Internet of Things based on the Firebase Real-Time Database”,IEEE International Conference on Smart Manufacturing, Industrial & Logistic Engineering (SMILE) February 2018.

[5] Ei Ei Thu and Nwe New, “Model Driven Development of Mobile Applications Using Drools Knowledge-based Rule”,IEEE 15th International Conference on Software Engineering Research, Management and Application (SERA) June 2017.Stand. Abbrev., in press.

[6] Kwanghoon Choi, Naeyon Bak and Byeong-Mo Chang “Smart Block : A Visual Programming Environtment for Smart Things”, Computer Engineering and Systems (ICCES), 2017 12th International Conference

[7] Alexandra Poulovassilis, George Papamarkos, and Peter T Wood,”Event-Condition-Action Rule Languages for the Semantic Web”, International Conference on Extending Database Technology 2006.

[8] Astrid Gloria Pepita and Tutun Juhana, “User Interface, Creation Retrieval of User Heatlh Information with Google Firebase, and Delivery of Automatic Emergency SMS for Ambient Assisted Living System : Monitoring of Eldery Condition using Smart Device”,4th International Conference on Wireless and Telematics (ICWT) 2018.