

SMART DOOR LOCKS BASED ON INTERNET of THINGS CONCEPT WITH MOBILE BACKEND AS A SERVICE

Muhammad Izzuddin Mahali

Jurusan Pendidikan Teknik Elektronika dan Informatika FT UNY

Email : izzudin@uny.ac.id

ABSTRACT

Smart Door Locks System Based on Internet of Things Concept with Mobile Backend as a Service is the smart door locks development are supported by cloud computing technology as data storage. Methods of development that exist in this system is to combine technology (1) ESP8266 (2) Firebase and (3) Android Apps. Firebase is one of cloud-based services that are used to store data online. ESP8266 able to communicate with firebase well to process the read and write data directly using internet communication channels. Android application used to direct communication to the firebase for monitoring and controlling simultaneously to the system. Android app can control / monitoring Indirectly to ESP8266 module via Firebase that serves as a mobile Backend as a Service to apply the rules of the Internet of Things.

Keywords : ESP8266, IoT, Backend Service, Firebase,

ABSTRAK

Sistem *Smart Door Locks Based on Internet of Things Concept with Mobile Backend as a Service* merupakan pengembangan kunci pintu pintar yang didukung dengan teknologi cloud computing sebagai penyimpan data. Metode pengembangan yang ada pada sistem ini adalah menggabungkan teknologi (1) ESP8266 (2) Firebase dan (3) Aplikasi Android. Firebase merupakan salah satu layanan berbasis *cloud computing* yang digunakan untuk menyimpan data secara online. ESP8266 mampu melakukan komunikasi dengan firebase baik untuk proses membaca data maupun menulis data secara langsung menggunakan jalur komunikasi internet. Aplikasi android digunakan untuk monitoring sekaligus mengendalikan sistem dengan cara melakukan komunikasi langsung ke firebase. Aplikasi android dapat mengontrol/monitoring modul ESP8266 melalui Firebase yang berfungsi sebagai mobile Backend as a Service dengan menerapkan kaidah Internet of Things.

Kata kunci : ESP8266, IoT, Backend Service, Firebase

PENDAHULUAN

Perkembangan teknologi bidang elektronik saat ini sudah sampai pada generasi Internet of Things. Internet of Things (IoT) adalah jaringan benda-benda fisik atau "things" yang tertanam (embedded) dengan elektronik, perangkat lunak, sensor dan konektivitas untuk memungkinkannya mencapai nilai yang lebih besar dengan

bertukar data dengan produsen, operator dan / atau perangkat lain yang terhubung. Setiap hal yang unik diidentifikasi melalui sistem komputasi tertanam (embedded) tetapi mampu beroperasi dalam infrastruktur internet yang ada. Menurut hasil penelitian dari Juniper Research terdapat pertumbuhan perangkat IoT 3 kalilipat antara tahun 2016 s/d 2021. Menurut hasil penelitian ini

perkiraan jumlah peralatan IoT yang terhubung ke internet baik itu device, sensor maupun aktuator di mencapai lebih dari 46 billion sampai dengan tahun 2021[1].

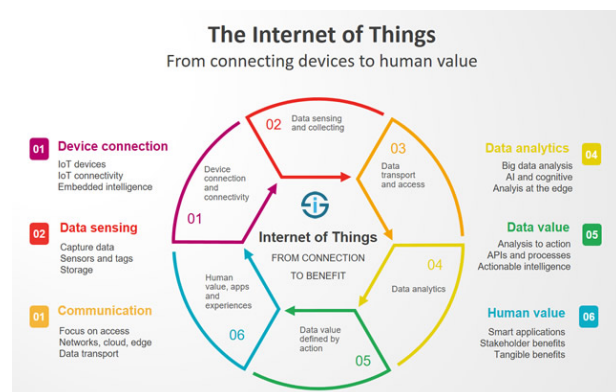
Pada tiga tahun terakhir Cloud Computing mengalami perkembangan cukup pesat, hingga merambah ke ranah *mobile communication*. Berbagai-macam bentuk service seperti *Infrastructure as a Service (IaaS)*, *Platform as a Service (Paas)* dan *Software as a Service (SaaS)* merubah sudut pandang pengguna layanan Teknologi Informasi (IT) dari *hardware oriented* menjadi *service oriented*. Penggabungan dari 3 layanan IaaS, Paas dan SaaS saat ini mempengaruhi perkembangan *mobile communication*. Berdasar pada riset yang dilakukan oleh Michael Facemire, 12 April 2012 seorang *Professional Application Development & Delivery* pada lembaga Forrester, mengungkapkan bahwa permintaan aplikasi berbasis mobile menjadi sangat tinggi mulai dari end user sampai dengan perusahaan besar. Cloud Database merupakan salah satu layanan yang dimiliki Mobile Backend as a service.

Kunci merupakan salah satu dari alat pengamanan rumah. Sistem pengaman rumah yang baik sangat di pengaruhi oleh kualitas kunci. Berbagai jenis kunci yang digunakan untuk sistem pengaman rumah mulai dari kunci manual sampai dengan kunci otomatis dengan bantuan energi listrik. Smart Door Locks Based On Internet Of Things (IoT) Concept with Mobile Backend as a Service merupakan penelitian pengembangan kunci elektronik berbasis internet of things. ESP8266 modul digunakan sebagai jembatan komunikasi antara smart lock dengan Cloud Database dalam mBaaS. Cloud Database diigunakan untuk menyimpan data pendukung sistem. Smart Door Locks dapat dikontrol dan dimonitoring secara jarak jauh

dengan menggunakan smart phone berbasis Android.

INTERNET OF THING

Internet of Things (IOT) adalah jaringan benda-benda fisik atau "things" yang tertanam (embedded) dengan sistem elektronik, perangkat lunak, sensor dan konektivitas dan memungkinkannya untuk mencapai nilai yang lebih besar dan layanan dengan saling bertukar data antar produsen, operator dan / atau perangkat lain yang saling terhubung [3]. Setiap hal yang unik diidentifikasi melalui sistem komputasi tertanam (embedded) tetapi mampu beroperasi dalam infrastruktur internet yang ada. "A Things" pada Internet of Things dapat didefinisikan sebagai subjek misalkan orang dengan monitor implant jantung, hewan peternakan dengan transponder biochip, sebuah mobil yang telah dilengkapi built-in sensor untuk memperingatkan pengemudi ketika tekanan ban rendah. Sejauh ini, IoT paling erat hubungannya dengan komunikasi machine-to-machine (M2M) di bidang manufaktur dan listrik, perminyakan, dan gas. Produk dibangun dengan kemampuan komunikasi M2M yang sering disebut dengan sistem cerdas atau "smart". (contoh: smart label, smart meter, smart grid sensor)[4].



Gambar 1. Internet of Things(<http://www.iscoop.eu>)

Untuk membangun sistem Internet of Things membutuhkan komponen yaitu device

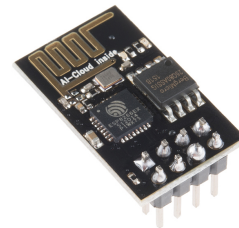
connection dan Data sensing. Selain komponen untuk membangun sistem IoT kemampuan berkomunikasi antara sistem juga dibutuhkan dalam IoT. Untuk menyimpan serta melakukan Data analytics dari data hasilhasil akusisi Data Sensing digunakan server database. Komponen terakhir adalah pemanfaatan dari komunikasi yang dijalankan terus menerus antara device connection dengan data sensing yang mampu menyimpan serta melakukan data analytics dan digunakan untuk membantu manusia dalam hal tertentu.

Istilah "Internet of Things" pertama kali didokumentasikan oleh seorang visioner Inggris, Kevin Ashton, pada tahun 1999. Biasanya, IOT diharapkan untuk menawarkan konektivitas canggih perangkat, sistem, dan jasa yang melampaui mesin-ke-mesin komunikasi (M2M) dan mencakup berbagai protokol, domain, dan aplikasi. Interkoneksi perangkat ini tertanam (termasuk benda pintar), diharapkan untuk mengantarkan otomatisasi dalam hampir semua bidang, sementara juga memungkinkan aplikasi canggih seperti Smart Grid.

MODUL ESP8266

ESP8266 adalah sebuah embedded chip yang di desain untuk komunikasi berbasis wifi. Chip ini memiliki output serial TTL dan GPIO. ESP8266 dapat digunakan secara sendiri (Standalone) maupun digabungkan dengan pengendali lainnya seperti mikrokontroler. ESP8266 memiliki kemampuan untuk networking yang lengkap dan menyatu baik sebagai client maupun sebagai Access Point. Firmware yang dimiliki ESP8266 begitu banyak, dapat juga sebuah chip ESP8266 diprogram dengan tujuan khusus sesuai dengan kebutuhan sebagai contoh kemampuan untuk berkomunikasi dengan web yang menggunakan port HTTPS.

Chip ESP8266 disempurnakan oleh Tensilica's seri L106 Diamond dengan prosesor 32-bit. Ada 3 cara menggunakan ESP8266 : sebagai wifi access menggunakan AT command, dimana biasanya dimanfaatkan oleh Arduino untuk koneksi wifi, sebagai sistem yang berdiri sendiri menggunakan NodeMCU dan menggunakan bahasa LUA, sebagai sistem yang berdiri sendiri dengan menggunakan Arduino IDE yang sudah bisa terhubung dengan ESP8266. ESP 8266 dapat bertindak sebagai client ke suatu wifi router, sehingga saat konfigurasi dibutuhkan setting nama access pointnya dan juga passwordnya, selain itu ESP8266 dapat digunakan sebagai Access Point dimana ESP8266 dapat menerima akses wifi.



Gambar 2. ESP8266-1 AI Cloud

Fitur yang dimiliki oleh ESP8266-1 sebagai berikut :

- Frekuensi wifi 802.11 b/g/n
- Prosesor 32 - bit MCU
- 10 - bit ADC
- TCP/ IP protocol stack
- TR switch, LNA , power amplifier dan jaringan
- PLL , regulator , dan unit manajemen daya
- Mendukung keragaman antena
- WiFi 2.4 GHz , mendukung WPA / WPA2
- Dukungan STA mode operasi /AP / STA + AP
- Dukungan Smart link Fungsi untuk kedua perangkat Android dan iOS
- SDIO 2.0 , (H) SPI , UART , I2C , I2S , IR Remote Control , PWM , GPIO
- STBC , 1x1 MIMO , 2x1 MIMO
- A - MPDU & A - MSDU agregasi & 0.4s guard interval

Mobile Backend as s Service (mBaaS)

Cloud Computing merupakan tren dunia teknologi pada saat ini, ekspansi dari

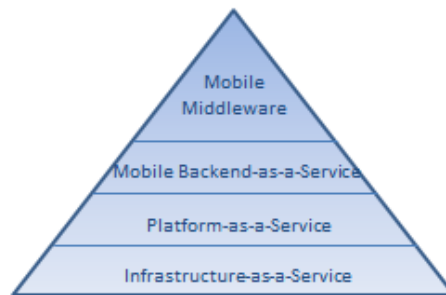
cloud computing merambah pada *mobile communication*. Berbagai macam layanan yang diberikan dalam *Cloud Computing* memiliki berbagai macam bentuk *service* diantaranya *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS) dan *Software as a Service* (SaaS). Saat ini *service* yang disediakan merubah orientasi pada produk layanan *Information Teknologi* (IT) dari yang bersifat *hardware oriented* menjadi *service oriented*. Pada saat ini bidang *mobile communication* menjadi leader teknologi yang memanfaatkan ketiga bentuk *service* tersebut.

Mobile Backend as a Service (mBaaS) merupakan salah satu layanan *cloud computing* yang memungkinkan seorang *mobile app developer* melakukan integrasi antara *database*, *cloud storage*, *push notification*, *management user*, *API* (*Application Program Interface*) dan *SDK* (*Software Development Kit*). Dukungan integrasi pada banyak *platform* disediakan dalam mBaaS. Kemudahan dalam pengelolaan *user database*, *file management*, *social networking integration*, *location services*, dan mengelola *load balancer* dari *traffic* yang masuk kedalam aplikasi *mobile* ditawarkan dalam mBaaS.

Piramida *framework* pengembangan aplikasi berbasis *mobile* layer paling bawah adalah (IaaS) kemudian layer PaaS, setelah itu layer mBaaS hingga pada *Mobile Middleware*. mBaaS memberikan layer layanan antara penyedia layanan PaaS dengan ranah solusi *end-to-end* yang dihasilkan dari *platform* aplikasi berbasis *mobile*.

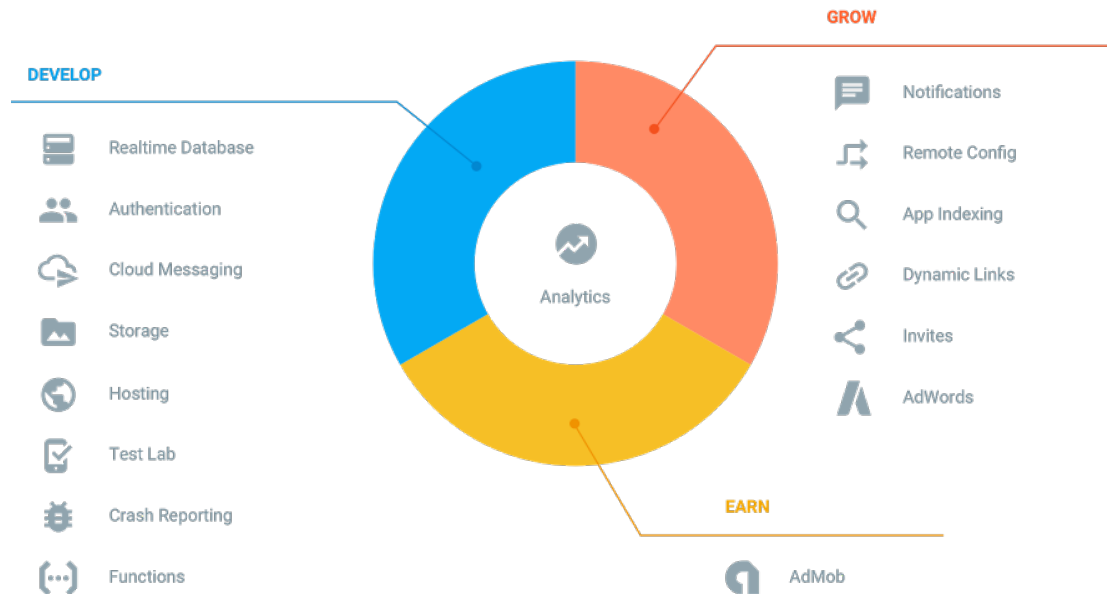
Atas dasar tuntutan teknologi dalam pengembangan aplikasi berbasis *mobile* saat ini, penyedia layanan mBaaS memberikan layanan berupa : (1) Fasilitas penyimpanan data berbasis *cloud computing* (2)

Pembentukan antar muka pemrograman aplikasi yang bersifat *Representational State Transfer* (RESTfull) dan otomatis untuk menyediakan akses baca/tulis terhadap data (3) Optimasi pengaksesan data menggunakan *JavaScript Object Nitation* (JSON) (4) Fasilitas otentikasi pengguna dan (5) Data analytics.



Gambar 3. Piramida *framework* pengembangan aplikasi *mobile*

Firestore merupakan salah satu dari sejumlah penyedia layanan mBaaS. Firestore mengalami perubahan besar-besaran sejak Mei 2016 dengan versi terakhirnya diberi nama Firestore 3.0. Dibandingkan dengan versi terdahulu saat ini dengan Firestore 3.0 memberikan *service* layanan yang lebih lengkap dari layanan terdahulu yang hanya memberikan layanan *authentication service* dan *real-time database* saja. Saat ini lebih kurang 15 layanan yang disediakan Firestore 3.0 seperti ditunjukkan pada gambar 3. Firestore memberikan layanan untuk *Service Develop* pada saat pengembangan aplikasi yaitu (1) *Realtime Database* (2) *Authentication* (3) *Cloud Messaging* (4) *Storage* (5) *Hosting* (6) *Test Lab* (7) *Crash Reporting* dan (7) *Cloud Functions*. Selain layanan untuk pengembang aplikasi pada sisi *service* untuk end user aplikasi Firestore 3.0 memberikan *service* berupa (1) *Notification* (2) *Remote Config* (3) *App Indexing* (4) *Dynamic Link* (5) *Invites* dan (7) *Adword*.



Gambar 4. Firebase Service (<http://firebase.com>)

Firebase 3.0 juga menyediakan layanan untuk berbagai merchant dalam menawarkan produk dalam service layanan AdMob. Layanan Data Analytics untuk analisa data juga disematkan dalam firebase 3.0 seiring dengan tren analisa Big Data saat ini.

ANDROID

Android merupakan sebuah sistem yang digunakan untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi [7]. Android mempunyai sifat open source sehingga siapa saja orang dapat ikut berperan dalam mengembangkan sistem ini. Pada awal berdirinya android yang merupakan perusahaan dengan nama Android inc. Selanjutnya pada perkembangannya Google membeli android inc yang bergerak pada pendatang baru piranti lunak untuk ponsel / smart phone. Pada awal perkembangan android dibentuklah Open Handset Alliance yang merupakan forum konsorsium 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi termasuk didalamnya

Google, HTC, Intel, Motorola, Qualcomm, T-Mobile dan Nvidia.

Android mempunyai empat karakteristik yaitu (1) terbuka, (2) Semua aplikasi (3) Memecahkan hambatan pada aplikasi dan (4) Memiliki banyak library dan tool pendukung yang banyak. Secara garis besar arsitektur android tersusun atas (1) Application (2) Application Framework (3) Libraries (4) Android Runtime dan (5) Linux Kernel. Komponen yang menjadi dasar aplikasi android antara lain : (1) Activities (2) Services (3) Content providers (4) Broadcast receivers [10].



Gambar 5. Arsitektur Android (<http://developer.android.com>)



Gambar 6. Diagram blok sistem

METODE PENELITIAN

Penelitian ini dilaksanakan di Laboratorium Jurusan Pendidikan Teknik Elektronika dan Informatika. Berikut ini adalah gambaran sistem yang dibuat seperti pada gambar 6. ESP8266 sebagai modul wifi juga berperan sebagai peralatan kontrol yang dapat memutar motor yang terdapat pada pintu dan berfungsi sebagai kunci. ESP8266 melakukan komunikasi baik baca dan tulis dengan Firebase realtime database secara online dengan terhubung ke access point. Smart phone android yang sudah terinstall aplikasi dapat melakukan pengontrolan dengan merubah status kunci. Aplikasi android dapat melakukan pembacaan dan penulisan realtime database secara online. Apabila salah satu dari device melakukan update data, secara otomatis smartphone lainnya yang sudah terinstall aplikasi juga akan mendapatkan update terakhir keadaan kondisi dari database. Firebase yang berfungsi sebagai Mobile Backend as Service memiliki banyak fasilitas dan diatur oleh App Engine (gambar 6).

HASIL DAN PEMBAHASAN

a. mBaaS Firebase Configuration

Penggunaan firebase sebagai backend database membutuhkan

konfigurasi. Konfigurasi untuk akses database dilakukan untuk 2 jenis platform yang berbeda. Yang pertama adalah untuk platform Embedded sistem dan yang kedua adalah untuk platform android. Konfigurasi realtime database untuk embedded sistem yang dibutuhkan adalah address host dan key database. Dalam pengaturan rule database juga diatur semua device dapat melakukan read/write data. Pengaturan ini dilakukan untuk komunikasi antara ESP8266 dengan firebase baik membaca data maupun menulis data. Data konfigurasi untuk penelitian ini adalah

```
HOST "smart-door-lock-df527.firebaseio.com"
Database key
"bWIBYTvPXUGjxQNXDflz6DjToDQU27ZAzt5U2aIL"
```

Sedangkan untuk rule database sebagai berikut :

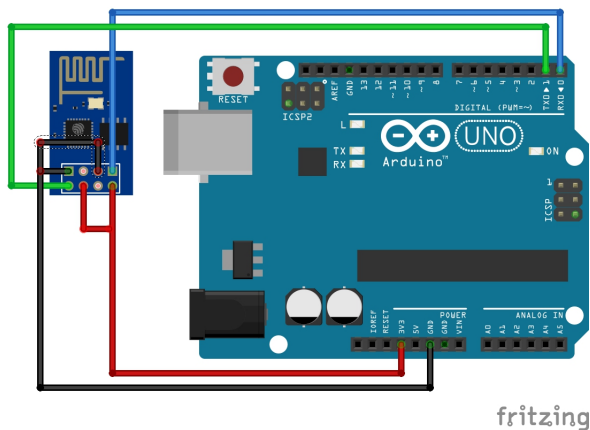
```
{
  "rules": {
    ".read": true,
    ".write": true
  }
}
```

Konfigurasi setting untuk platform android sudah diberikan file konfigurasi dalam bentuk satu buah file yang bisa diunduh dari firebase.com setelah nama package aplikasi yang akan

dibuat ditambahkan dalam sistem tersebut. Nama file yang harus dimasukkan dalam package aplikasi supaya dapat mengakses realtime

b. Perancangan Program ESP8266-1

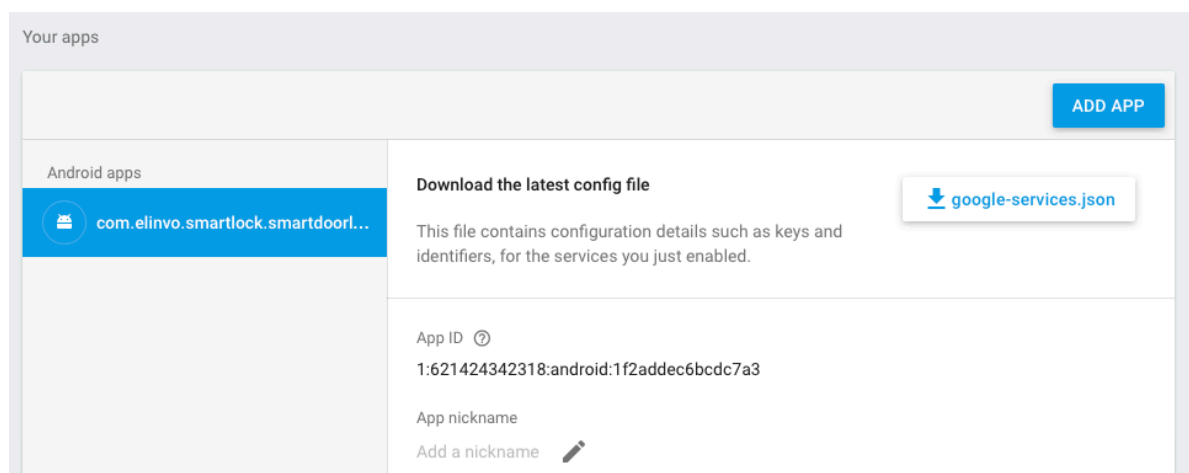
ESP8266 merupakan embedded chip yang dapat di program dengan menggunakan arduino IDE dan juga bahasa LUA. Pada penelitian ini pada proses programming ke ESP8266 menggunakan bantuan board Arduino UNO tanpa chip ATMEGA sebagai interface dengan IDE. Berikut ini adalah skematik diagram pada saat programming ESP8266



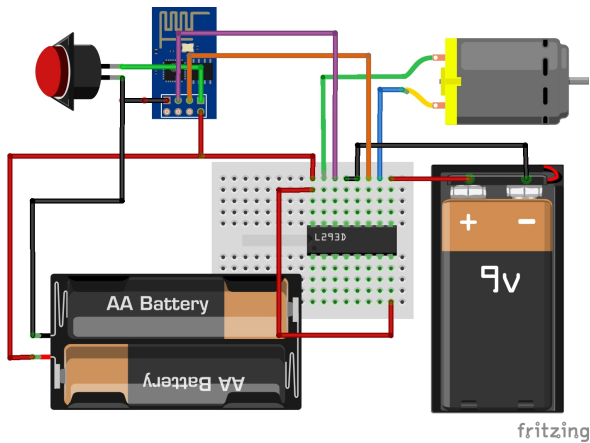
Gambar 7. Diagram ESP8266 pada saat programming

database di firebase adalah `google-services.json`. Berikut ini tampilan project setting dalam firebase sekaligus lokasi untuk mengunduh file konfigurasi

Pada penelitian ini dibutuhkan 2 buah port GPIO untuk mengendalikan motor dan 1 buah port input untuk tombol. ESP8266 secara umum hanya memiliki 2 buah GPIO untuk general input/output. Pemanfaatan port serial Rx sebagai input data tombol digunakan sebagai alternatif tambahan port. Dalam program yang dimasukkan dalam ESP8266 *serial communication* hanya diaktifkan untuk port TX saja, kemudian port RX diseting sebagai port input dan bertipe PULLUP. Pengendali motor dengan menggunakan IC L293 membutuhkan 2 port untuk mengendalikan sehingga motor dapat bergerak 2 arah (searah dan berkebalikan arah jarum jam) karena keadaan Smart Door Lock terdapat 2 proses membuka dan mengunci. IC L293 hanya dimanfaatkan 1 buah jembatan karena motor yang digunakan hanya satu buah. Gambar 8 adalah gambar rangkaian sistem secara keseluruhan.



Gambar 8. konfigurasi Firebase untuk Android SDK



Gambar 9. Rangkaian sistem

Berikut ini adalah *source code* smart door lock dengan menggunakan Arduino IDE

```
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#define FIREBASE_HOST "smart-door-lock-
df527.firebaseio.com"
#define FIREBASE_AUTH
"bWIBYTvPKUGjxQNXDflz6DjToDQU27ZAzt5U2aIL"
#define WIFI_SSID "HTC6545LVW"
#define WIFI_PASSWORD "meteor1"

const int motorPin1 = 0; //pin 15 L293
const int motorPin2 = 2; //pin 10 L293
const int tombol = 3; //pin tombol

int val = 0;

void setup() {
  //Serial hanya pin TX yang digunakan pin
  Serial.begin(115200, SERIAL_8N1, SERIAL_TX_ONLY);
  pinMode (motorPin1, OUTPUT);
  pinMode (motorPin2, OUTPUT);
  pinMode (tombol, INPUT_PULLUP); //GPIO RX untuk

  digitalWrite (motorPin1, LOW);
  digitalWrite (motorPin2, LOW);

  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

long n = 0;
int lockstate;
int lockstatebefore = 0;

void loop() {
  val = digitalRead(tombol);
  if (val == LOW)
  {
    val = digitalRead(tombol);
    if (val == HIGH)
    {
      lockstate = Firebase.getInt("lock");
      if (lockstate == 1){
        Firebase.setInt("lock", 0);
        bukakunci();
      }
    }
  }
  else {
    Firebase.setInt("lock", 1);
    kunci();
  }
}

Serial.println(Firebase.getInt("lock"));
lockstate = Firebase.getInt("lock");

if (Firebase.failed()) {
  Serial.print("pushing /logs failed:");
  Serial.println(Firebase.error());
  return;
}

if (lockstate != lockstatebefore){
  if (lockstate == 1){
    Serial.println("Terkunci");
    kunci();
  }
  else {
    Serial.println("Terbuka");
    bukakunci()
  }
}
lockstatebefore = lockstate

String name = Firebase.pushInt("logs", n++);
// handle error
if (Firebase.failed()) {
  Serial.print("pushing /logs failed:");
  Serial.println(Firebase.error());
  return;
}
Serial.print("pushed: /logs/");
Serial.println(name);

delay(1000);
}

//Membuka Kunci
void bukakunci(){
  digitalWrite (motorPin1, HIGH);
  digitalWrite (motorPin2, LOW);
  delay (3000);
  digitalWrite (motorPin1, LOW);
  digitalWrite (motorPin2, LOW);
}

//Mengunci
void kunci(){
  digitalWrite (motorPin1, LOW);
  digitalWrite (motorPin2, HIGH);
  delay (3000);
  digitalWrite (motorPin1, LOW);
  digitalWrite (motorPin2, LOW);
}
```

```
} else {
  Firebase.setInt("lock", 1);
  kunci();
}
}

Serial.println(Firebase.getInt("lock"));
lockstate = Firebase.getInt("lock");

if (Firebase.failed()) {
  Serial.print("pushing /logs failed:");
  Serial.println(Firebase.error());
  return;
}

if (lockstate != lockstatebefore){
  if (lockstate == 1){
    Serial.println("Terkunci");
    kunci();
  }
  else {
    Serial.println("Terbuka");
    bukakunci()
  }
}
lockstatebefore = lockstate

String name = Firebase.pushInt("logs", n++);
// handle error
if (Firebase.failed()) {
  Serial.print("pushing /logs failed:");
  Serial.println(Firebase.error());
  return;
}
Serial.print("pushed: /logs/");
Serial.println(name);

delay(1000);
}

//Membuka Kunci
void bukakunci(){
  digitalWrite (motorPin1, HIGH);
  digitalWrite (motorPin2, LOW);
  delay (3000);
  digitalWrite (motorPin1, LOW);
  digitalWrite (motorPin2, LOW);
}

//Mengunci
void kunci(){
  digitalWrite (motorPin1, LOW);
  digitalWrite (motorPin2, HIGH);
  delay (3000);
  digitalWrite (motorPin1, LOW);
  digitalWrite (motorPin2, LOW);
}
```

Dalam program diatas secara garis besar terbagi menjadi 4 bagian (1) Menghubungkan ESP8266-1 ke akses point (2) Pengecekan input tombol dengan *anti bounch support* (3) Akses read/write data pada firebase dan (4) Kendali motor untuk membuka dan mengunci.

c. Perancangan Aplikasi Android

Pada tahap pengembangan aplikasi android yang mampu berkomunikasi dengan firebase terdapat beberapa point penting mengenai

konfigurasi yang dilakukan. Setelah konfigurasi pada sisi mBaaS firebase dilakukan selanjutnya pada sisi pengembangan aplikasi dilakukan proses memasukkan file konfigurasi firebase ke dalam package. Adapun nama file konfigurasi yang diberikan dari firebase adalah google-service.json. Ketika android mengakses layanan Firebase perlu ditambahkan library *service* pada build.gradle `'com.google.gms:google-services:3.0.0'`. Selain itu diperlukan library firebase realtime database `'com.google.firebase:firebase-database:9.6.1'` yang berfungsi untuk mengakses realtime database pada mBaaS Firebase [9]. Pada proses *inserting data* source code yang digunakan untuk menuliskan data dalam firebase realtime database adalah sebagai berikut :

```
mFirebaseInstance = FirebaseDatabase.getInstance();
mFirebaseInstance.getReference("lock").setValue(1);
```

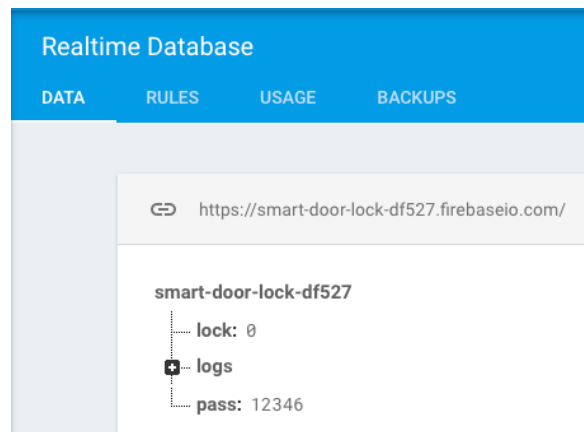
Terdapat perbedaan yang besar antara membaca dan menulis firebase realtime database. Dalam proses membaca realtime database dilakukan menggunakan listener, berikut ini adalah *sourcecode* untuk membaca data dari firebase realtime database.

```
ValueEventListener getPass = new
ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot
dataSnapshot) {
        codepass = (Long) dataSnapshot.getValue();
    }
    @Override
    public void onCancelled(DatabaseError
databaseError) {
    }
};
mFirebaseInstance.getReference("pass")
.addValueEventListener(getPass);
```

d. Implementasi dan pengujian

Implementasi pada mBaaS firebase realtime database terlebih dahulu dibuat filed data sebagai basis penyimpanan data. Pada penelitian ini field yang dibutuhkan ada 2 yaitu fields

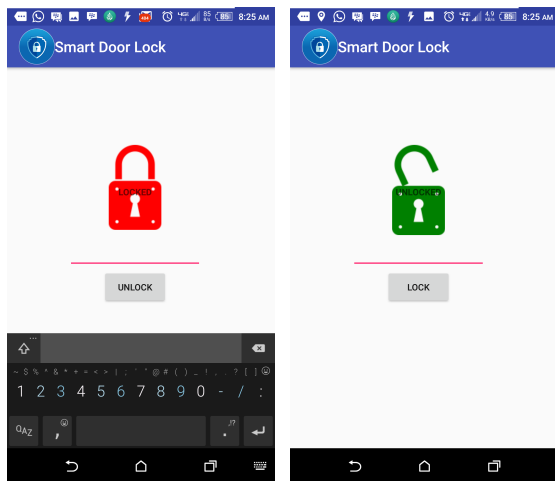
pertama adalah “lock” dengan type data long, field ini digunakan untuk merepresentasikan keadaan smart door. Nilai 1 merepresentasikan smart door dalam keadaan terkunci dan nilai 0 merepresentasikan keadaan terbuka. Field yang kedua yaitu “pass” dengan type data long, field ini digunakan untuk menyimpan password akses pada sistem. Pada penerapan untuk aplikasi Android sebelum melakukan kendali smart door lock terlebih dahulu memasukkan password dan harus sesuai dengan isi field dalam database. Apabila password tidak sama smart door lock tidak akan menjalankan perintah Lock ataupun Unlock.



Gambar 10. Data Structure Realtime Database pada Firebase

Berikut ini adalah implementasi aplikasi android Smart Door Lock. Terdapat 3 buah komponen utama dalam aplikasi ini yaitu : (1) Icon Lock/Unlock (2) Password Field dan (3) Button. Icon dalam bentuk gembok dalam aplikasi ini akan merepresentasikan keadaan dalam database filed “lock”, apabila field tersebut bernilai 1 yang merepresentasikan keadaan pintu terkunci maka tampilan dalam aplikasi akan seperti pada Gambar 9 (a) dengan icon gembok terkunci dan berwarna merah. Apabila filed “lock” bernilai 0 maka

dalam aplikasi akan menampilkan icon gembok terbuka dan berwarna hijau. Tampilan icon akan berubah secara otomatis ketika database dalam firebase berubah, sebagai contoh ketika terdapat 4 buah smartphone android yang menggunakan aplikasi Smart Door Lock, apabila salah satu dari 4 smart phone tersebut mengunci ataupun membuka kunci smart door lock maka ke 3 smartphone yang lainnya akan menampilkan keadaan terakhir dari smart door lock.

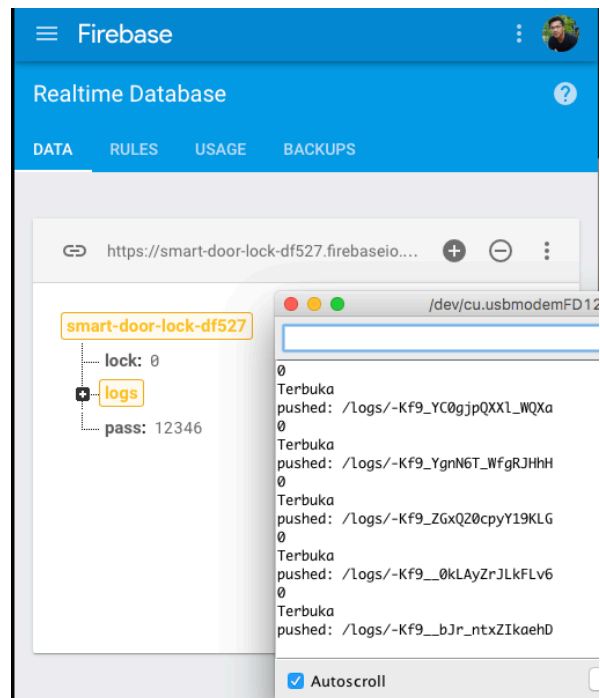


Gambar 11. Aplikasi android Smart Door Locks
(a) Lock dan (b) Unlock

Pada saat implementasi komunikasi yang terjadi antara ESP8266 dengan firebase realtime database dilakukan secara online ditunjukkan pada gambar 12. Dalam database field “lock” akan selalu dibaca oleh ESP8266. ESP8266 akan selalu mengikuti nilai yang terdapat dalam field “lock”. Warna kuning dalam firebase realtime database menandakan bawah ada update data dalam field “logs”. Pada proses ini ESP8266 mengirimkan data logs pada sistem database.

Pada implementasi sistem, tombol digunakan sebagai kontrol manual untuk proses membuka dan mengunci Smart Door Lock. Pada saat tombol ditekan

sistem Smart Door Lock akan melakukan pengecekan keadaan terakhir selanjutnya sistem akan merubah keadaan yang semula terbuka menjadi terkunci atau sebaliknya. Sistem ini menggunakan akses 1 tombol untuk membuka dan mengunci pintu.



Gambar 12. Response Arduino dengan Firebase

SIMPULAN

Pengembangan sistem pada penelitian ini mampu menghubungkan embedded sistem ESP8266 yang berfungsi sebagai kontrol dengan firebase realtime database yang berperan sebagai mobile Backend as a Service untuk menyimpan data. Modul ESP8266 mampu membaca keadaan database dan melaksanakan perintah sesuai dengan data yang ada. Selain itu Modul ESP8266 juga mampu mengirimkan data ke Firebase realtime database. Aplikasi berbasis Android juga digunakan sebagai alat untuk membuka ataupun mengunci smart door locks melalui jalur komunikasi internet. Seluruh device yang sudah terinstal aplikasi ini akan melakukan sinkronisasi dengan ESP8266 melalui firebase realtime database sehingga akan mendapatkan keadaan yang sama apabila smart door locks pada keadaan

terkunci maupun terbuka. Oleh karena itu penelitian Smart Door Lock sudah dapat dikategorikan dalam sistem berbasis Internet of Things.

DAFTAR RUJUKAN

Aulia Faqih Rifa'i, 2016, Sistem Pendeteksi Dan Monitoring Kebocoran GAS (Liquefied Petroleum Gas) Berbasis Internet of Things, JISKa, Vol. 1, No. 1, Mei, 2016, Pp. 5 – 13 ISSN 2527-5836

(tambah)Berkah I Santosa. 2012, Ebook : Santoso, Mobile Backend as a Service, <http://www.cloudindonesia.or.id/mobile-backend-services.html>. diakses agustus 2016

(rename)Developer Android, Application Fundamental, <https://developer.android.com/guide/components/fundamentals.html> , diakses agustus 2016

(rename)Google Firebase, Add Firebase to Your Android Project, <https://firebase.google.com/docs/android> , diakses Agustus 2016

Gupta ,Neetu; Mandal, Ritabarta dan Chaddha, Vinary. 2016. Internet of Things based Boor Locking System-An Architecture. International Science Press. IJCTA , pp. 385 = 390

(rename)I-SCOOP,-, Internet of Things – IoT guide with definitions example, trends dan use cases. <http://www.i-scoop.eu/internet-of-things-guide/>. diakses agustus 2016

Internet of Things; privacy and security in the connected world, FTC Staff Report, January 2015.

Juniper Research, “‘Internet of Things’ Connected Devices to Almost Triple to over 38 Billion Units by 2020,” press

release, July 28, 2015. diakses Agustus 2016

Juniper Research, Retail Spend on ‘Internet of Things’ to reach \$2.5Bn by 2020, <http://www.juniperresearch.com/press/press-releases/retail-spend-on-iot-to-reach-2-5bn-by-2020> diakses Agustus 2016

Madakam, Somayya. 2015. *Internet of Things: Smart Things International*. Journal of Future Computer and Communication, Vol. 4, No. 4,

Nazrudin Safaat. 2012 , *Android : Pemrograman Aplikasi Mobile Smartphone dan Tablet PC (Edisi Revisi)*, Bandung, Informatika,

(tambah)Onno W. Purbo, 2015, Membuat Sendiri Internet of Things untuk Pemula, Khususnya di Indonesia, <https://infokomputer.grid.id/2015/10/fitur/membuat-sendiri-internet-of-things-untuk-pemula-khususnya-di-indonesia/>. diakses agustus 2016.

R. Das, "Stage 1 - Introduction to the Internet of Things: What, Why and How - CodeProject:," Grasshopper.iics, 26 Oktober 2014. [Online]. Available: <http://www.codeproject.com/Articles/832492/Stage-Introduction-to-the-Internet-of-Things-Wha>. [Accessed 20 Agustus 2016].

Yu Liu, dkk. 2015. *Combination of Cloud Computing and Internet of Things (IOT) in Medical Monitoring Systems*. International Journal of Hybrid Information Technology. Vol.8, No.12, pp. 367 – 376.

Zanella, Andrea dan Vangelista, Lorenzo, 2014, Internet of Thing for Smart Cities, Internet of Things Journal, vol. 1, no 1, Feb 2014.