

PENENTUAN KOORDINAT GLOBAL POSITIONING SYSTEM (GPS) RAMBU DILARANG PARKIR UNTUK MEREKAM PELANGGARAN BERBASIS INTERNET OF THINK (IOT)

Ega Pratama Alfriyani¹, Slamet Winardi²

^{1,2)} Program Studi Sistem Komputer Universitas Narotama Surabaya

Jl. Arief Rahman Hakim No. 51 Surabaya

Email : ega.pratama@mhs.fasilkom.narotama.ac.id,

slamet.winardi@narotama.ac.id

ABSTRACT

The Industrial revolution 4.0 in the field of technology that is Internet of think (IoT) then to overcome the problem of traffic signs violations, need for a tool that can help police to monitor users the way through the internet with Android applications. And in 2018 Rizal makes an application of digital STNK based on Internet of think (IoT) which uses Wemos D1 mini as a microcontroller. Lack of this tool is the absence of violation record features and Global positioning system (GPS). The purpose of this study is to create a tool that can record violations of parking sign and to geting latitude and longtitude coordinates using the GPS function on the smartphone. To obtain data in the form of a police number, date, and time, the following devices are used: wemos d1 mini as a server, wemos d1 mini as a client, codular creator to create android applications.

Keyword: *Android, Coordinate, Internet, Violations, Wemos*

ABSTRAK

Revolusi Industri 4.0 di bidang teknologi yaitu Internet of Think (IoT) untuk mengatasi masalah pelanggaran rambu lalu lintas tersebut, diperlukan suatu alat yang dapat membantu polisi untuk memantau pengguna jalan melalui internet dengan aplikasi Android. Dan pada tahun 2018, Rizal membuat aplikasi STNK digital berbasis Internet of think (IoT) yang menggunakan mikrokontroler Wemos D1 mini. Kekurangan dari alat ini adalah tidak adanya fitur rekam pelanggaran dan Global Positioning System (GPS). Tujuan dari penelitian ini adalah membuat alat yang dapat merekam pelanggaran rambu dilarang parkir dan mendapatkan koordinat lintang dan bujur menggunakan fungsi GPS pada smartphone. Untuk memperoleh data berupa nomor polisi, tanggal, dan waktu digunakan perangkat-perangkat berikut: wemos d1 mini sebagai server, wemos d1 mini sebagai klien, pembuat kode untuk membuat aplikasi android.

Kata kunci : *Android, Koordinat, Internet, Pelanggaran, Wemos*

1. PENDAHULUAN

Perilaku masyarakat pengguna jalan terutama pengendara bermotor baik mobil atau motor yang cenderung tidak peduli terhadap peraturan lalu lintas disekitarnya. Di Indonesia seperti yang ditulis dalam jurnal (Machsus *et al.*, 2014). Berdasarkan data Perhubungan Darat Dalam Angka (PDDA,2012), disebutkan jumlah kecelakaan lalu lintas jalan pada tahun 2011 mencapai 109.776 kejadian. Kendaraan yang terlibat 239.257 kendaraan dengan total korban 176.763 orang, yang terdiri dari: korban meninggal sejumlah 31.185 orang, 36.767 orang mengalami luka berat dan 108.811 orang mengalami luka ringan.

Indonesia telah memasuki revolusi Industri 4.0 dibidang teknologi yaitu IoT (*Internet of things*) (Junaidi, 2016) maka untuk mengatasi masalah yang telah dipaparkan diatas, perlunya suatu alat yang bisa membantu polisi untuk mengawasi atau memperingatkan pengguna jalan melalui internet yang berkolaborasi dengan pemrograman *Android*.

Merancang sebuah sistem kendali pada lampu dengan *multicontrol* berupa : *bluetooth*, suara, cahaya, *timer*, *website* dan SMS sebagai notifikasi status lampu. Sistem yang dirancang

mampu mengendalikan lampu dari jarak dekat maupun jauh karena menggunakan jaringan global.(Herpendi *et al.*, 2018)

Penggunaan chips ESP8266 untuk membuat identifikasi kendaraan yang berisi data kendaraan dan pemilik. identitas kendaraan (SSID) berupa nomor polisi setiap kendaraan akan ditanamkan pada chip ESP8266 bersifat unik. Selanjutnya melakukan pemasangan chip ESP8266 yang sesuai pada kendaraan dengan identitas pada chip tersebut. Bila terdapat *access point* maka chip akan terhubung dan memberikan SSID ke *access point* dan diteruskan ke *server*. membuat STNK digital menggunakan modul arduino yang kemudian dilakukan sinkronisasi antara data digital dengan nomor polisi. Membuat aplikasi karcis berbasis *Android* atau aplikasi IOS (Limantara *et al.*, 2017).

Pembuatan alat atau sistem untuk mengirimkan data kendaraan ke *smartphone* menggunakan mikrontroler arduino *Wemos DI Mini ESP8266* sebagai chip pada STNK Digital. App Inventor sebagai *web* aplikasi untuk membuat aplikasi *smartphone* khusus sistem operasi *Android* untuk menampilkan data dari XAMPP sebagai *web server* (Rizal *et al.*, 2018).

Pemantauan kelembaban tanah dan tingkat air tanah menggunakan sensor ultrasonik untuk memprediksi kegagalan lereng ketika ada curah hujan tinggi. Mereka telah menggunakan metode pemantauan kelembaban tanah dengan *wireless* untuk menentukan metode penempatan sensor yang tepat. Juga menggunakan sensor DHT11 untuk mengetahui suhu, kelembaban, curah hujan dan cahaya, yang dipasang di bidang pertanian. Sensor-sensor ini akan mengumpulkan parameter lingkungan. Lalu data penginderaan dimitigasi ke *cloud* melalui *gateway IOT (Internet of Things)* mikrokontroler yang digunakan disini ialah *wemos* (Rukhmode *et al.*, 2017).

Dengan alat yang canggih dapat membantu polisi untuk mengawasi pengguna jalan yaitu sepeda motor dan mobil secara *wireless* dengan Aplikasi *Android* yang akan menentukan koordinat *latitude longitude* dimana pengguna jalan melanggar rambu tersebut, alat ini menggunakan modul *Wemos D1 Mini* sebagai *hardware*, aplikasi *android* sebagai *software*.

2. METODE

Tahapan pengerjaan merupakan sesuatu yang harus dilakukan dalam penelitian. Beberapa tahapan yang diterapkan meliputi pengumpulan data,

perancangan *hardware*, perancangan *software*, implementasi sistem, hasil dan pembahasan.

a. Metode Pengambilan Data

Proses pengambilan data yang dibutuhkan dalam penelitian ini.

1) Studi Literatur

Informasi dari berbagai sumber seperti internet, artikel, dan skripsi sebagai penunjang penelitian. Agar penelitian ini dapat menghasilkan produk yang lebih berkembang dari penelitian sebelumnya.

2) Survei Lokasi Secara Langsung

Dalam pengambilan data koordinat *Latitude* dan *Longitude* penelitian ini, penulis melakukan pengambilan secara langsung pada 6 titik dimana rambu dilarang parkir terpasang. Data dituliskan dalam bentuk tabel 1.

Tabel 1 Hasil Pengumpulan Data

| Alamat Lokasi (Surabaya) | <i>Latitude</i> | <i>Longitude</i> |
|-----------------------------------|-----------------|------------------|
| 1 | 2 | 3 |
| Depan Indomart, Jl. Embong Malang | -7.258878 | 112.733955 |
| Jl. Embong Malang No. 10-42 | -7.260306 | 112.737279 |
| Depan Hotel JW Marriot | -7.259439 | 112.735156 |
| JL. Basuk Rahmat No. 08 | -7.266646 | 112.740743 |
| JL. Basuk Rahmat No. 78-62 | -7.267427 | 112.741118 |
| JL. Basuk Rahmat No. 111-141 | -7.270615 | 112.741622 |

b. Perancangan *Hardware*

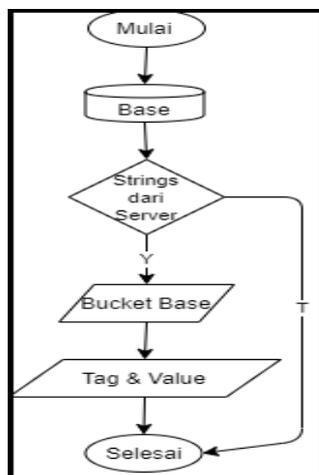
Merancang *Hardware* untuk membangun penelitian ini, menggunakan dua buah mikrokontroler *Wemos D1 mini*. *Wemos D1 mini* sebagai *server* dan juga *Wemos D1 mini* sebagai *client*. disini *client* berisi nomor polisi dari kendaraan(motor/mobil). Dari *server* dilanjutkan menuju *database* lalu ke aplikasi *android* untuk merekam pelanggaran dilarang parkir.

c. Perancangan *Software*

Untuk menjelaskan alur sistem yang tepat dibutuhkan sebuah *flowchart*, agar memudahkan dalam memahami cara kerja program yang sedang dibangun. Dalam merancang *software* digunakan 2 *software* aplikasi *web* yaitu *Kodular* dan *Firebase*.

1) Membangun *Database Firebase*

Untuk menampung data yang dikirim dari *server*, diperlukan tempat atau *base*. Sebagaimana gambar 1.

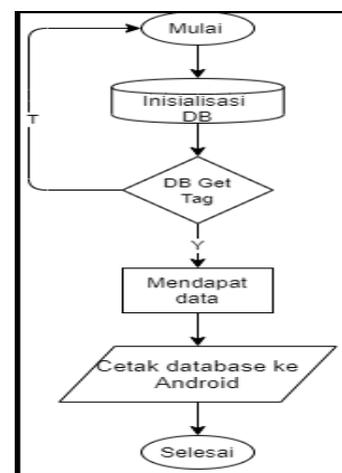


Gambar 1 *Flowchart* base di *firebase*

Base sebelumnya sudah dibuat saat membuat *real-time database* jika data telah diterima dari *server* maka *database* membuat *bucket* yang berisi *tag & value*.

2) *Firebase* ke aplikasi *android*

Proses berikutnya pengiriman data dari *database* ke *android*. Sebagaimana gambar 2.



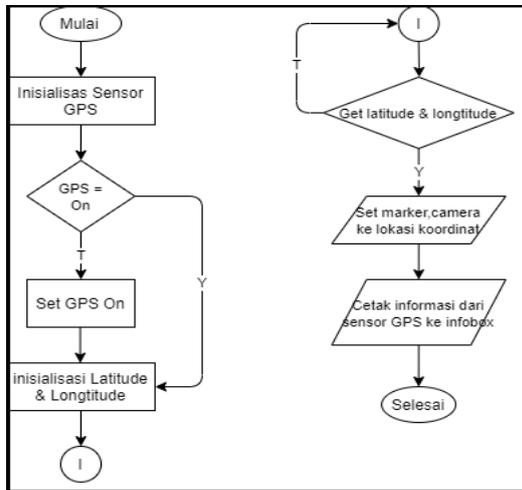
Gambar 2 *Flowchart* mengirim *database* ke *android*

Inisialisasi *database* saat awal aplikasi berjalan apabila *database* berhasil ditemukan maka aplikasi berhasil mendapat data, label menampilkan data dari masing-masing *tag* dan *value*.

3) Inisialisasi Koordinat *GPS*

Ketika semua data dari *database* telah siap begitu juga dengan koordinat dari sensor *GPS* telah berhasil didapatkan, rancangan *software* dan *hardware* bisa di implementasikan

sesuai dengan tahapan *flowchart* sebagaimana gambar 3.

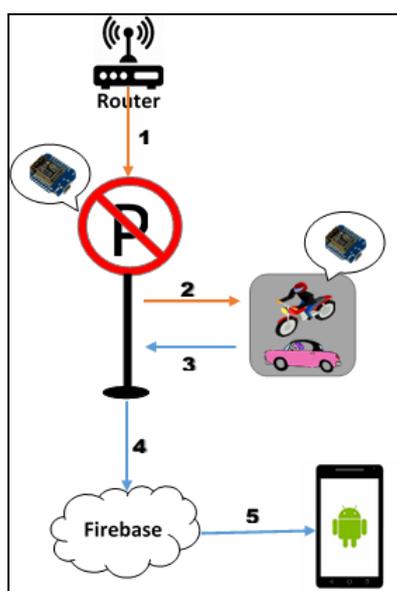


Gambar 3 *Flowchart* inisialisasi GPS

Penelitian ini mengacu pada koordinat *latitude* dan *longitude*, maka diperlukan sensor *GPS* dari hp pengguna.

d. Implementasi Sistem

Tahap ini merupakan tahap dimana keseluruhan sistem siap untuk dioperasikan. Sebagaimana gambar 4 adalah *flowchart* implementasi sistem.



Gambar 4 Implementasi Sistem

Proses bekerjanya alat ini sesuai *flowchart* gambar 4 adalah sebagai berikut:

- 1) *Wemos D1 mini* sebagai yang berada di rambu dilarang parkir menghubungkan ke *router* sebagai *access point* utama, maka terbentuklah *server*.
- 2) *Server* telah memiliki *access point* sendiri, kemudian menghubungkan ke *Wemos D1 mini* sebagai *client* yang memiliki data nomor polisi kendaraan. Jika telah saling terhubung (*server* dan *client*) terhubung,
- 3) Data nomor polisi dikirim kembali ke *server*, dan *server* mendapat nomor polisi,
- 4) *Server* yang telah menerima data nomor polisi, menyimpan data berupa nomor polisi, jam, dan tanggal ke *firebase* didalam *real-time database*.

Data kemudian dikirim dan ditampilkan ke aplikasi *Android* berserta dengan posisi koordinat *latitude* dan *longitude*.

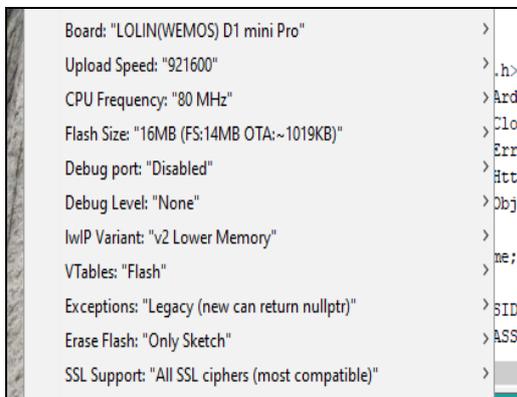
3. HASIL DAN PEMBAHASAN

Pada bagian ini *server* dan *client* pada masing-masing *wemos d1 mini* adalah salah satu tahapan yang penting agar program dalam penelitian ini dapat

berjalan sesuai dengan tahapan perencanaan yang sudah dilakukan.

a. Wemos D1 Mini sebagai Server

Membuat *wemos d1 mini* menjadi *server* dimana *wemos* menggunakan *software* arduino, *setting library* untuk *board* menggunakan LOLIN (WEMOS) D1 mini beserta *library* lain yang dibutuhkan. *Set library wemos* sebagaimana gambar 5.



Gambar 5 Set Library Wemos

Akses poin untuk sumber koneksi untuk *server*, `WIFI_SSID = "GKPB MDC JOMBANG"`, dan `WIFI_PASSWORD = "GKPBMDJCJOMBANG"` sebagai sumber koneksi utama. Kemudian set AP (Akses Poin/Hotspot) untuk membuat akses poin baru dari *server* `ssid = "New AP"`, `apassword = "wemoslololn"`. Satu bagian yang penting adalah set *IPAddress* (*Internet Protocol Address*). Untuk *IPAddress* di *setting* dengan `ip (192,168,11,4)`. Sebagaimana gambar 6.

```
const char* WIFI_SSID = "GKPB MDC JOMBANG";
const char* WIFI_PASSWORD = "GKPBMDJCJOMBANG";

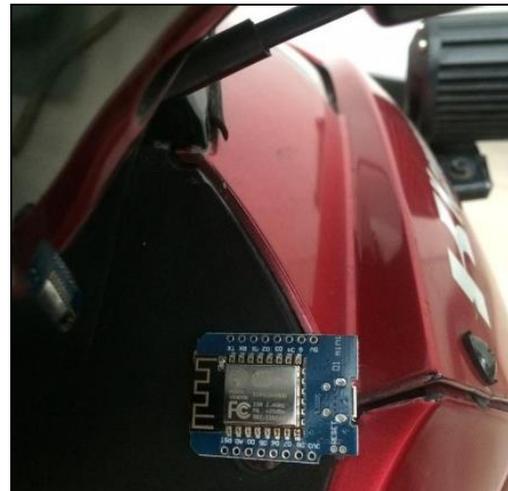
const char* ssid = "New AP"; // Nama AP/Hotspot
const char* apassword = "wemoslololn"; // Password AP/Hotspot

IPAddress ip(192,168,11,4);
IPAddress gateway(192,168,11,1);
IPAddress subnet(255,255,255,0);
```

Gambar 6 Set Akses Poin Server

b. Wemos D1 Mini sebagai Client

Tahap berikutnya adalah membangun *client*. Dimana *wemos* berada di kendaraan (motor/mobil). Pada motor, *wemos* diletakkan pada *dashboard* di bawah stang kemudi sebagaimana gambar 7.



Gambar 7 Wemos sebagai client pada motor

Pada mobil, *wemos* diletakkan di dashboard dalam mobil diatas *audio* sebagaimana gambar 8.



Gambar 8 Wemos sebagai client pada mobil

Alat ini bekerja bila *client* berhasil terhubung dengan *server*. AP *client* harus terhubung dengan Akses poin yang dimiliki *Server*. *Host* pada *client* sama dengan *IPAddress* *server*, apabila tidak sama maka tidak bisa saling terhubung. Set `WIFI_SSID = "New AP"`; `WIFI_PASSWORD = "wemoslolin"`; `host = "192.168.11.4"`. Sebagaimana gambar 9.

```
const char* WIFI_SSID = "New AP";
const char* WIFI_PASSWORD = "wemoslolin";
const char* host = "192.168.11.4";
//ini client
WiFiClient client;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.println();
  Serial.println();
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
```

Gambar 9 Menghubungkan *server* ke *client*

c. Pengujian *Server* dan *Client*

Untuk memastikan alat bekerja dibutuhkan pengujian hubungan antara *Server* dan *Client*. Jika `WIFI_SSID` dan `WIFI_PASSWORD` benar maka akan muncul tulisan "Wifi Tersambung", lalu *server* akan meng-set *IPAddress* pada serial monitor. *Serial monitor server* sebagaimana gambar 10.

```
connecting.....
Wifi Tersambung
IP address:
192.168.43.115
Server dijalankan
192.168.43.115
```

Gambar 10 Serial Monitor *Server*

Wemos D1 mini sebagai *server* telah berhasil berjalan. Dalam *serial monitor* tercetak "connecting." *server* sedang melakukan koneksi dengan *access point* utama yaitu wifi, jika *server* tersambung maka *serial monitor* menampilkan *IPAddress* dan *server* siap dijalankan. Dilanjutkan pengujian *client* yang telah tersambung dengan *server*. *Serial monitor client* sebagaimana gambar 11.

```
Updated
Wifi Tersambung
IP address (AP): 192.168.11.103
Nomor Polisi anda: S1676KL
```

Gambar 11 Serial Monitor *Client*

Pada serial monitor proses menghubungkan `WIFI_SSID` dan `WIFI_PASSWORD` akses poin *server* New AP, serial monitor mencetak "Connecting...". Kemudian "Wifi Tersambung", dan set *IPAddress*. *Client* menampilkan Nomor Polisi. Disini Pengujian hubungan antara *Server* dan *Client* berhasil.

d. Sent Nomor Polisi ke server

Pengiriman nomor polisi dilakukan *client* sebagai *chip* yang ada di kendaraan kepada *server*. Arduino *Client* kirim ke *server* sebagaimana gambar 12.

```
//data noNopol
String noNopol = "S1676KL";
Serial.print("Nomor Polisi anda: "); Serial.println(noNopol);

//konek ke server da mengirim data noPolisi menggunakan URL
if(client.connect(host,80)) {
  String url = "/update?value=";
  url += String(noNopol);
  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
  "Host: " + host + "\r\n" + "Connection: keep-alive\r\n\r\n");
  delay(10);
  //membaca semua line yang di respon dan di print ke serial
  Serial.println("Response: ");
  while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
  }
}
```

Gambar 12 Arduino *Client* kirim ke *server*

Gambar 12 menjelaskan pada *client* mengirim nomor polisi melalui *url* dengan tipe data *String* pada *arduino IDE* lalu mencetaknya pada serial monitor menjadi seperti gambar 13.

```
Updated
Wifi Tersambung
IP address (AP): 192.168.11.103
Nomor Polisi anda: S1676KL
Response:
HTTP/1.1 200 OK
Content-Type: text/plain
Content-Length: 7
Connection: close

Updated
 Autoscroll  Show timestamp
```

Gambar 13 Serial Monitor *Client* 2

Server menerima data nomor polisi, mencetaknya pada serial monitor. *Server* telah memiliki data Nomor Polisi. *Serial Monitor Server 2* sebagaimana gambar 14.

```
connecting.....
Wifi Tersambung
IP address:
192.168.43.115
Server dijalankan
192.168.43.115
S1676KL
S1676KL
 Autoscroll  Show timestamp
```

Gambar 14 Serial Monitor *Server 2*

e. Pengujian *Server* ke *Firebase*

Setup *Host* and *Auth* sesuai dengan *base* yang dibuat agar data dari arduino bisa ditampung, jikalau tidak ada *base* untuk data, data tidak bisa disimpan. Setup *Host & Auth* sebagaimana gambar 15.

```
// Firebase
#define FIREBASE_HOST "testwemos-e05d5.firebaseio.com"
#define FIREBASE_AUTH "6Uz3fjLU2kF4ACZIp81XlucZeofMeUR32jnlig8j"
```

Gambar 15 Setup *Host & Auth*

Set *String* untuk disimpan di *firebase* sebagai *real-time database*. Set *NoPol* untuk menyimpan nomor polisi yang telah diterima server lalu dikirim ke *firebase*. *Get NoPol* sebagaimana gambar 16.

```
void pegangIndex() {
  server.send(200,"text/plain",String(noNopol)); // ngirim tulisan pertama
}

void pegangUpdate() {
  //dikirim dengan url
  noNopol = server.arg("value");
  Serial.println(noNopol);
  server.send(200,"text/plain","Updated");
}
```

Gambar 16 Get *NoPol*

`myLocalTime.setLocation("Asia /Jakarta")`, untuk menentukan *Time Zone*, `Firebase.setString("jam",myLocalTime.dateTime("H:i:s"))`; untuk mengirim ke *firebase* lalu membuat objek dengan tag "jam" dan *value* yaitu jam, menit, dan detik. Proses yang sama untuk "tgl", "/pelanggaran/jam", "/pelanggaran/tanggal". *Set String* arduino IDE sebagaimana gambar 17.

```
void pegangUpdate() {
  //dikirim dengan url
  noNopol = server.arg("value");
  Serial.println(noNopol);
  server.send(200, "text/plain", "Updated");
  Firebase.setString("/pelanggaran/jam", myLocalTime.dateTime("H.i.s"));
  Firebase.setString("/pelanggaran/tanggal", myLocalTime.dateTime("d.m.Y"));
  Firebase.setString("/pelanggaran/Nopol", noNopol);
}
```

Gambar 17 Set String arduino IDE

Firebase berhasil membaca data yang dikirim dari Arduino dan menulis data tersebut secara *Real time*. Hasil *Real-time Database* seperti gambar 18. Pengujian *server* ke *firebase* berhasil.



Gambar 18 Database

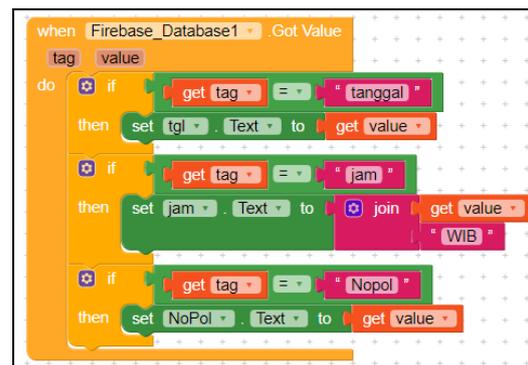
f. Firebase ke Aplikasi Android

Data yang ditampilkan pada aplikasi diambil dari *firebase*. Aplikasi

dibuat dengan aplikasi *web Kodular Creator*.

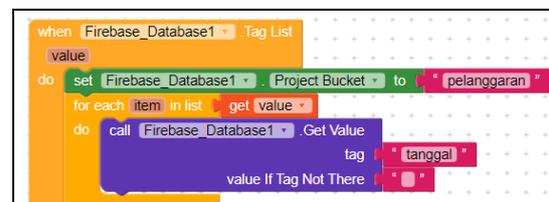
1) Inisialisasi Database

Melakukan inisialisasi data dari *firebase*, jika tersedia maka aplikasi bisa lanjut bekerja. *Get value* sebagaimana gambar 19.



Gambar 19 Get Value

Gambar 19 menunjukkan cara aplikasi *android* memanggil *value* dari *firebase*. *Tag list* sebagaimana gambar 20.



Gambar 20 Tag List

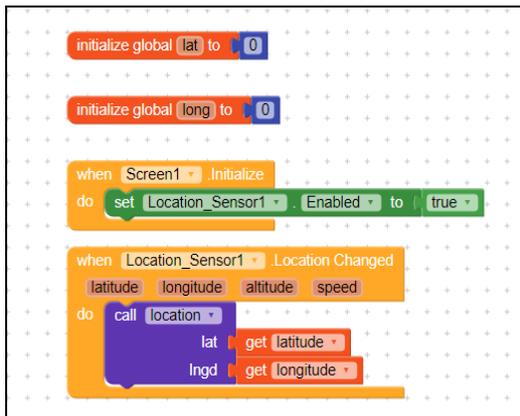
Menerima data dan menampilkan ke *interface* aplikasi *android*. Tampilan Data *Firestore* di *Android* sebagaimana gambar 21.

| | |
|----------------|-------------|
| Nomor Polisi : | S1676KL |
| Jam : | 15.27.14WIB |
| Tanggal : | 19.06.2020 |

Gambar 21 Tampilan Data *Firestore* di *Android*

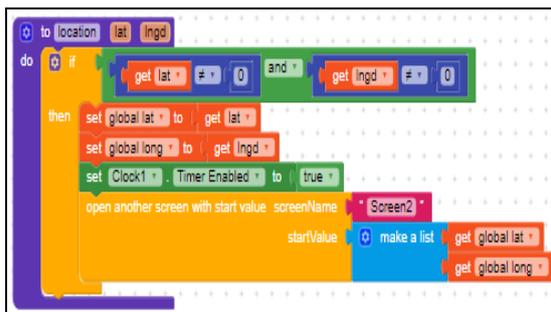
2) Inisialisasi Koordinat GPS

Menetapkan koordinat GPS berupa *latitude* dan *longtitude* dengan sensor yang dimiliki *smartphone android*. Inisialisasi GPS sebagaimana gambar 22.



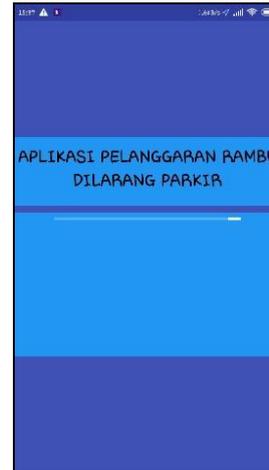
Gambar 22 Inisialisasi GPS

Block call location didapat dari *procedure* yang berisi perintah inisialisasi *latitude* dan *longtitude*. *Procedure location* sebagaimana gambar 23.



Gambar 23 Procedure location

Pada *procedure location*, bila telah mendapat *latitude* dan *longtitude*, memberikan perintah untuk membuka *screen2*. *Loading screen* sebagaimana gambar 24.

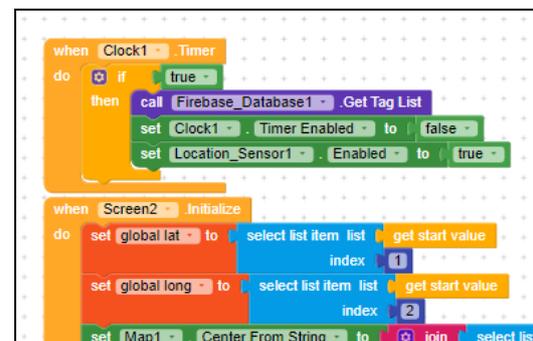


Gambar 24 Loading screen

Gambar 24 menunjukkan proses *loading screen* saat inisialisasi lokasi dan berpindah ke *screen2*.

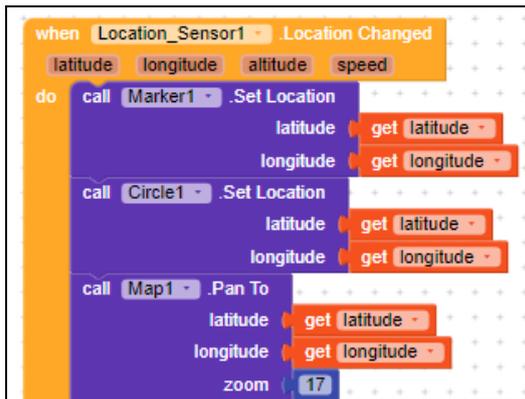
g. Aplikasi Perekam Pelanggaran

Aplikasi ini menerima data, merekam, menampilkannya, sehingga pengguna bisa melihat lokasi pelanggaran secara *real time*. *Screen2* sebagaimana gambar 25.



Gambar 25 Screen2

Proses yang terjadi di *screen2* ketika *screen* akan terbuka, *clock timer* memanggil i dan mengaktifkan sensor lokasi yang ada di *smartphone*. *Location changed* sebagaimana gambar 26.



Gambar 26 Location changed

Apabila *location changed*, posisi *marker*, *camera*, dan *circle* meletakkan posisinya di titik *latitude* dan *longitude* sesuai sensor lokasi. Hasil akhirnya adalah lokasi berhasil didapatkan dengan munculnya *infobox* yang berisi informasi tentang posisi *latitude*, *longitude*, dan alamat sesuai titik koordinat GPS. Hasil Aplikasi *Android* sebagaimana gambar 27.



Gambar 27 Hasil Aplikasi *Android*

Dengan adanya aplikasi ini diharapkan pengguna jalan bisa lebih tertib dalam mematuhi rambu dilarang parkir yang terpasang di sepanjang jalan.

4. SIMPULAN

Perancangan sistem dan pengujian telah dilakukan dengan tahapan-tahapan sesuai studi literature yang telah ditetapkan. Dengan menggunakan *wemos d1 mini* sebagai server dan client, *firebase* sebagai database, dan *kodular* sebagai android developernya. Alat dan aplikasi telah berjalan dengan baik, harapannya aplikasi ini dapat bekerja sesuai dengan tujuannya yaitu membantu polisi dalam mengawasi aturan rambu dilarang parkir dan menjadikan masyarakat bisa mentaati rambu tersebut.

DAFTAR PUSTAKA

Direktorat Jenderal Perhubungan Darat. (2012). *Perhubungan Darat Dalam Angka (PDDA)*.
 Herpendi, H., Julianto, V. and Hafizd, K. A. (2018) 'Perancangan Multicontrol Pada Lampu Berbasis Internet Of Think (IOT)', *Jurnal SAINTEKOM*, 8(2), p. 129. doi: 10.33020/saintekom.v8i2.65.
 Junaidi, A. (2016) 'Internet of Things , Sejarah , Teknologi Dan Penerapannya', *Jurnal Ilmiah*

- Teknologi Informasi*, I(AUGUST 2015), pp. 62–66.
- Limantara, Arthur Daniel Krisnawati, Lucia Desti Winardi, Slamet Mudjanarko, S. W. (2017) ‘Solusi Pengawasan Kebijakan Mengatasi Kemacetan Jalan dan Parkir Kota Berbasis Internet Cerdas’, *Seminar Nasional Teknologi dan Rekayasa Informasi*, (November), pp. 1–6.
- Machsus Sulistio, Harnen Wicaksono, A Djakfar, L. (2014) ‘Kajian tingkat kecelakaan lalu lintas di kota Surabaya’, *Prosiding Seminar Nasional Aplikasi Teknologi Prasarana Wilayah*, pp. 133–140.
- Rizal, Achmad Winardi, Slamet Supriyanto, Dadang Anindito, Benediktus Utomo, W. M. (2018) ‘Desain STNK Digital Dengan Chip Esp8266 Berbasis Internet Of Desain Stnk Digital Dengan Chip Esp8266 Berbasis Internet Of Things (Iot) Dalam Era Industri 4 . 0’, (October), pp. 0–8.
- Rukhmode, Sandip Vyavhare, Gayatri Banor, Sharda Narad, Abhilash Tugnayat, R. M. (2017) ‘IOT Based Agriculture Monitoring System Using Wemos’, (March), p. 2017. Available at: <http://www.ijritcc.org>.