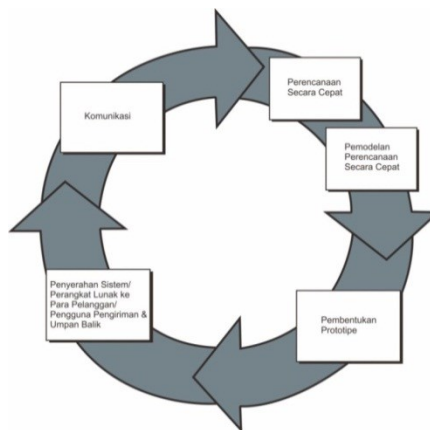


BAB III METODE PENELITIAN

3.1 Metode Pengembangan Perangkat Lunak

Dalam penelitian ini metode pengembangan perangkat lunak yang dapat digunakan untuk pembuatan “*Tracking Kereta Api Penumpang Jurusan Lampung Baturaja Berbasis Android*” yang telah dijelaskan pada bab 2.



Gambar 3.1 Metode *Prototype* (Pressman 2012)

3.2 Metode Pengumpulan Data

Pengumpulan data memegang peranan yang sangat penting dalam mendapatkan informasi dari penelitian yang dilakukan. Pengumpulan data harus dilakukan dengan metode pengumpulan data yang tepat. Data objektif dan relevan dengan pokok pembahasan menjadi indikator keberhasilan suatu penelitian. Dalam hal ini, peneliti menggunakan beberapa metode pengumpulan data antara lain :

1. Studi Pustaka

Pengumpulan data dengan cara membaca, mempelajari dan memanfaatkan buku-buku yang tersedia diperpustakaan, artikel, atau jurnal hasil penelitian orang lain yang digunakan sebagai literatur yang dapat mendukung di dalam penyusunan dan penulisan skripsi. Tujuan dari studi pustaka yaitu untuk menyusun dasar teori yang digunakan dalam melakukan penelitian.

2. Observasi

Observasi yang dilakukan peneliti adalah dalam bentuk pengamatan dan pencatatan langsung dan tidak langsung. Peneliti menggunakan observasi non partisipan, yaitu peneliti hanya mengamati keadaan objek secara langsung, tetapi peneliti tidak aktif dan tidak terlibat secara langsung.

3.3 Analisis Sistem

Perancangan sebelum di terjemahkan ke dalam komputer, dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan yang terjadi dan kebutuhan-kebutuhan yang diharapkan. Analisis yang akan menjalankan fungsi-fungsi yang bertujuan untuk memberikan layanan kepada penggunanya. Adapun kebutuhan dari perancangan yang akan diimplementasikan, yaitu perangkat keras dan perangkat lunak.

3.3.1 Analisis Kebutuhan Perangkat Lunak

Untuk membangun aplikasi pengenalan bagian tubuh manusia menggunakan tiga bahasa diperlukan setidaknya beberapa jenis perangkat lunak, yaitu perangkat lunak untuk membangun aplikasi. Maka dipilihlah perangkat lunak sebagai berikut :

a. Sistem Operasi

Digunakan untuk pembuatan aplikasi karena pembuatannya harus menggunakan sistem operasi untuk membuka *arduino* dan *MIT app inventor*. Untuk pembuatannya disini saya menggunakan sistem operasi windows 10.

b. *MIT App Inventor*

Digunakan untuk merancang *Syntax* atau merancang *Interface user* yang bertujuan untuk membangun sistem.

c. *Arduino*

Arduino IDE merupakan sebuah software untuk memprogram arduino. Kita sebut saja dengan bahasa pemrograman C for Arduino. Didalam arduino sendiri sudah terdapat IC mikrokontroler yang sudah ditanam program yang bernama Bootloader. *Arduino IDE* dibuat dari bahasa pemrograman JAVA yang dilengkapi dengan *library C/C++ (wiring)*, yang membuat operasi input/output lebih mudah.

3.3.2 Analisis Kebutuhan Perangkat Keras

Untuk menjalankan perangkat lunak diatas dibutuhkan perangkat keras dengan spesifikasi yang cukup, adapun spesifikasi minimum perangkat keras untuk menjalankan perangkat lunak diatas adalah sebagai berikut :

1. *Smartphone*

Digunakan untuk dapat menjalankan aplikasi yang sudah *terinstall* di *smartphone*. Adapun spesifikasi *smartphone* yang dapat dijalankan sebagai berikut :

Tabel 3.1 Tabel Spesifikasi *Smartphone*

Perangkat Keras	Spesifikasi
Sistem Operasi	<i>Android</i>
Versi Android	6.0 (Marshmallow)
RAM	4 GB
Memori	64 GB
Prosesor	Octa-core (2x2.0 GHz Cortex-A75 & 6x1.8 GHz Cortex-A55)

2. Laptop

Digunakan untuk membangun atau merancang aplikasi sesuai kebutuhan dalam menjalankan arduino dan *MIT app inventor*. Agar dalam pembuatan tidak ada kendala dan berjalan dengan baik. Berikut ini spesifikasi laptop untuk menjalankan arduino :

Tabel 3.2 Tabel Spesifikasi laptop

Perangkat Keras	Spesifikasi
Laptop	Acer Aspire V5-431
Prosesor	Intel Celeron CPU 1017U 1.6 GHz
RAM	6GB DDR-3
VGA	Intel HD <i>Graphics</i>
HDD	300 GB HDD

3. *NodeMCU ESP8266*

Rangkaian ini merupakan CPU Board yang berfungsi sebagai pengendali utama dari keseluruhan sistem. Board modul *embedded system* yang mempunyai *feature WiFi*. menggunakan chip ESP8266 dengan *firmware* berbasis Lua. Pada *NodeMCU* dilengkapi dengan *port Micro USB* yang berfungsi untuk pemrograman sekaligus *power supply*.

Tabel 3.3 Tabel Spesifikasi *NodeMCU ESP8266*

Perangkat Keras	Spesifikasi
Mikrokontroler / <i>Chip</i>	ESP8266-12E
Tegangan Input	3.3 ~ 5V
GPIO	13 Pin
Kanal PWM	10 Kanal
10 bit ADC Pin	1 Pin
<i>Flash Memory</i>	4 MB
<i>Clock Speed</i>	40/26/24 MHz
<i>WiFi</i>	IEEE 802.11 b/g/n
Frekuensi	2.4 GHz – 22.5 Ghz

<i>USB Port</i>	<i>Micro USB</i>
<i>USB Chip</i>	CH340G

4. *Module GPS Ublox Neo-6m*

GPS Ublox Neo-6m digunakan untuk untuk mendeteksi lokasi dengan menangkap proses sinyal dari navigasi. Modul ini kompatibel dengan APM2 dan APM2.5 dengan EEPROM terpadu yang dapat digunakan untuk menyimpan data konfigurasi. Antarmuka menggunakan serial TTL (RX/TX) yang dapat diakses dari mikrokontroler yang memiliki fungsi UART atau emulasi serial TTL (pada Arduino dapat menggunakan pustaka komunikasi serial / *serial communication library* yang sudah tersedia dalam paket Arduino IDE). Baud rate diset secara *default* di 9600 bps.

Tabel 3.4 Tabel Spesifikasi *Module GPS Ublox Neo-6m*

Perangkat Keras	Spesifikasi
Tipe penerima	50 kanal, GPS L1 <i>frekuensi</i> , C/A Code. SBAS: WAAS, EGNOS, MSAS
Sensitivitas penjejak & navigasi	161 dBm (reakuisisi dari <i>blank-spot</i> : -160 dBm)
Kecepatan pembaruan data	5 Hz
Akurasi penetapan lokasi <i>GPS</i>	2,5 meter (SBAS = 2m)

4. Alat-Alat Pendukung

a. Solder

Berfungsi Sebagai alat pendukung yang digunakan untuk memanaskan, timah patri yang digunakan untuk menyambung komponen-komponen elektronik.

b. Obeng

Digunakan untuk merapatkan atau mengunci mur komponen yang ada pada *NodeMUC ESP8266* dan module *GPS Neo-6m*.

3.4 Perancangan Sistem

Perancangan sistem tahap selanjutnya setelah melakukan analisa sistem. Mendapatkan gambaran dengan jelas apa yang dikerjakan sistem, maka dilanjutkan dengan pemikiran bagaimana membuat sistem tersebut. Terdapat 2 perancangan yaitu perancangan *prototype* dan perancangan rangkaian.

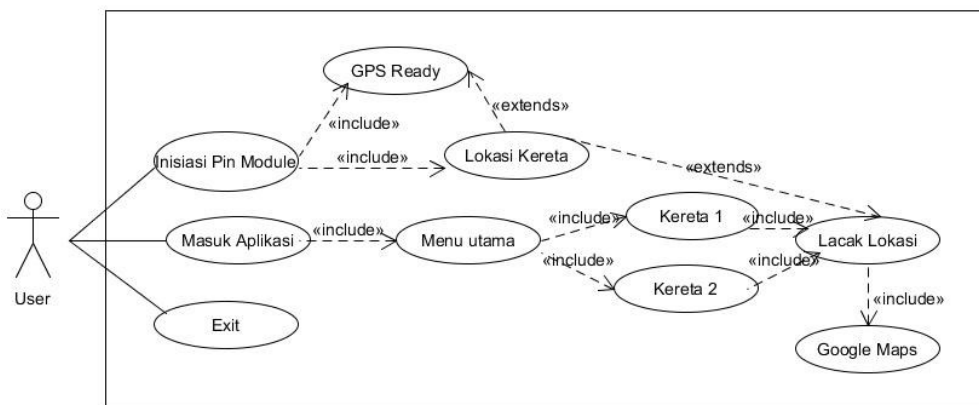
3.4.1 Perancangan *Prototype*

Perancangan *prototype* menggunakan *unified modeling language* (UML) yang terdiri dari *use case* diagram dan *activity* diagram.

3.4.1.1 *Use Case Diagram*

Use case diagram merupakan pemodelan untuk sistem informasi yang akan dibuat. Untuk deskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Pada sistem *tracking* kereta api, terdapat 1 aktor, yakni *user/* pengguna. Berikut gambar *Use Case Diagram* :

1. *Use Case Diagram User*



Gambar 3.2 Usecase Diagram.

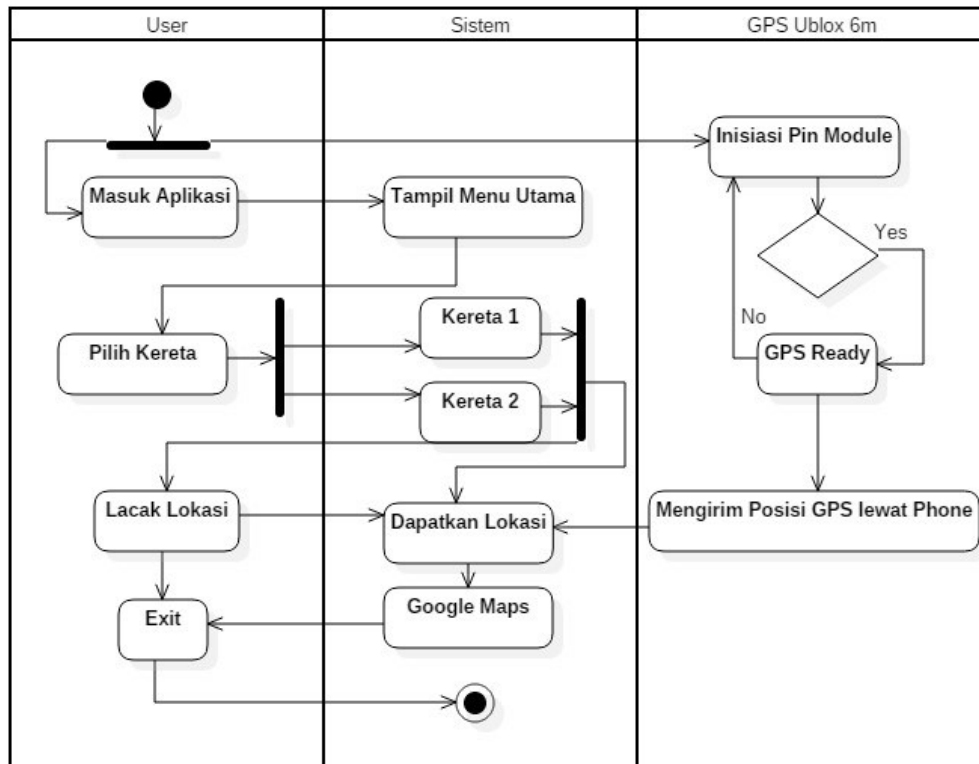
Pada sistem ini, aktor memiliki beberapa peran sebagai berikut:

- a. Inisiasi pin, Memasang pin yang ada pada *GPS ublox 6m* dengan *NodeMCU ESP8266*.
- b. *GPS ready*, menginformasikan bahwa *GPS* telah aktif.
- c. Masuk aplikasi, bertujuan masuk ke dalam aplikasi *smartphone*.
- d. Menu Utama, Fitur untuk menemukan posisi kereta.
- e. Kereta 1, menu kereta Lampung menuju Baturaja
- f. Kereta 2, menu kereta Baturaja menuju Lampung
- g. Posisi Kereta, menginformasikan posisi *GPS* menuju *smartphone* / aplikasi.
- h. Lacak Lokasi, memberikan informasi kepada pengguna akan sampai di stasuin tujuan dan mengingatkan barang-barang bawaan pengguna.
- i. *Google Maps*, bertujuan untuk mendapatkan lokasi dengan aplikasi *google maps*.
- j. Exit, Berfungsi untuk keluar dari aplikasi.

3.4.1.2 Activity Diagram

Activity diagram atau Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem. Berikut ini gambar *Activity Diagram* :

1. *Activity Diagram User*

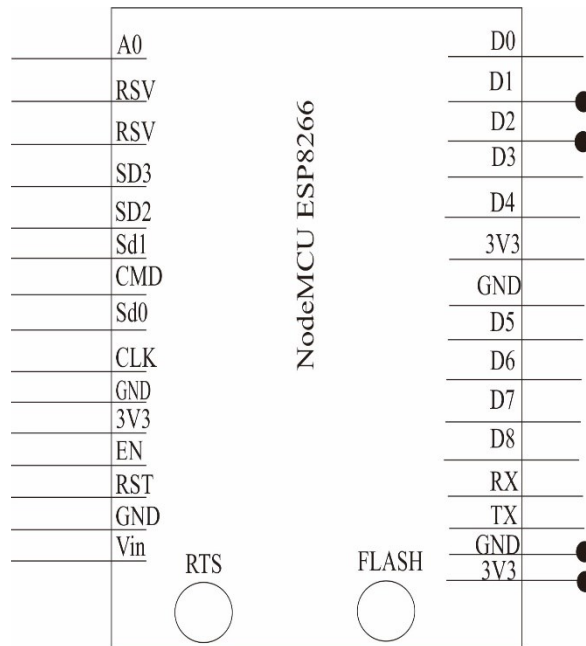


Gambar 3.3 *Activity Diagram User*

3.4.2 Perancangan Rangkaian

3.4.2.1 NodeMCU ESP8266

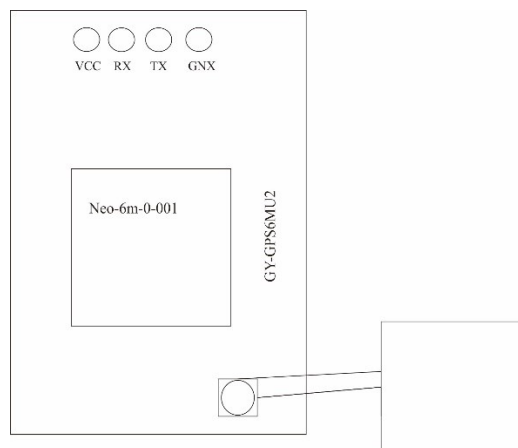
Rangkaian ini berfungsi sebagai pengendali dari sebuah sistem mikrokontroler atau dapat disebut sebagai otak. Board ini dilengkapi dengan port-port yang berhubungan dengan modul pendukung lain.



Gambar 3.4 Rancangan *NodeMCU EPS8266*

3.4.2.2 *Module GPS Ublox Neo 6-m*

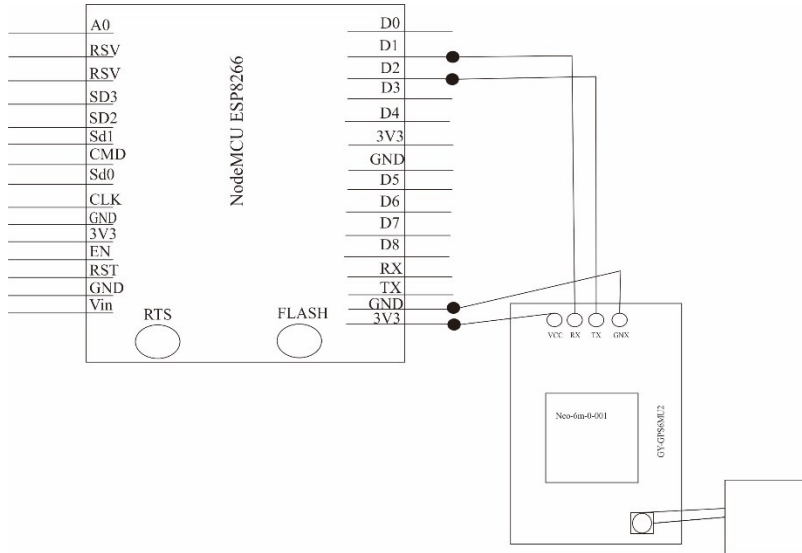
Sensor yang digunakan sebagai pendeteksi lokasi dan mengirim ke web. GPS ini cukup dapat diandalkan karena memiliki keakuratan yang cukup baik dan untuk menangkap sinyal dengan kuat.



Gambar 3.5 Rancangan *Module GPS Ublox 6-m*

3.4.2.3 Konfigurasi *NodeMCU ESP8266* dengan *GPS Ublox 6-m*

Agar terhubung sehingga bisa *tracking* lokasi kereta api, *NodeMCU EPS8266* harus dikonfigurasi dengan *GPS Ublox 6-m*. Seperti gambar berikut.



Gambar 3.6 Konfigurasi pin *NodeMCU ESP 8266* dengan *GPS Ublox 6-m*

Dengan rangkaian pin dari dan *module GPS Ublox 6-m*. Berikut tabel rangakian pin :

Tabel 3.5 Konfigurasi pin *NodeMCU ESP 8266* dengan *GPS Ublox 6-m*

No	NodeMCU ESP 8266	GPS Ublox 6-m
1	D1	RX
2	D2	TX
3	GND	GND
4	3V	VCC

3.4.3 Perancangan Perangkat Lunak

Pembuatan perangkat lunak atau pencodingan sistem harus mengutamakan cara kerja sistem yang efisien berikut tahap tahap dari sistem tersebut :

3.3.1.1 Perancangan Perangkat Lunak (*Arduino*)

Pengcodingan sistem menggunakan arduino. Berikut ini perancangan perangkat lunak atau pengcodingan :

```
void setup() {
  Serial.begin(115200);
  ss.begin(GPSBaud);
  delay(1000);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to ");
  Serial.print(WIFI_SSID);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("Connected to ");
  Serial.println(WIFI_SSID);
  Serial.print("IP Address is : ");
  Serial.println(WiFi.localIP());
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}
```

Gambar 3.7 Pemrograman *Connect Wifi (Arduino)*

Perancangan di atas adalah pemrograman untuk *connect wifi* agar bisa *mentracking* lokasi yang diinginkan.

Pemrograman ini untuk mendapatkan lokasi *tracking* dengan menggunakan *module GPS Ublox neo-6m*. *GPS* akan menerima titik koordinat seperti latitude dan longitude. Seperti gambar di bawah ini.

```

void loop() {
  while (ss.available() > 0)
    if (gps.encode(ss.read())){
float la= (gps.location.lat());
float lo= (gps.location.lng());
float kecepatan = gps.speed.kmph();
    if(kecepatan <=1){
        kecepatan=0;
    }
latitude = String(la,6);
longitude = String(lo,6);
    Serial.print("Location: ");
    if (gps.location.isvalid())
    {
        Serial.print(latitude);
        Serial.print (" ");
        Serial.println(longitude);
        Serial.print("Kecepatan: ");
        Serial.println(kecepatan);

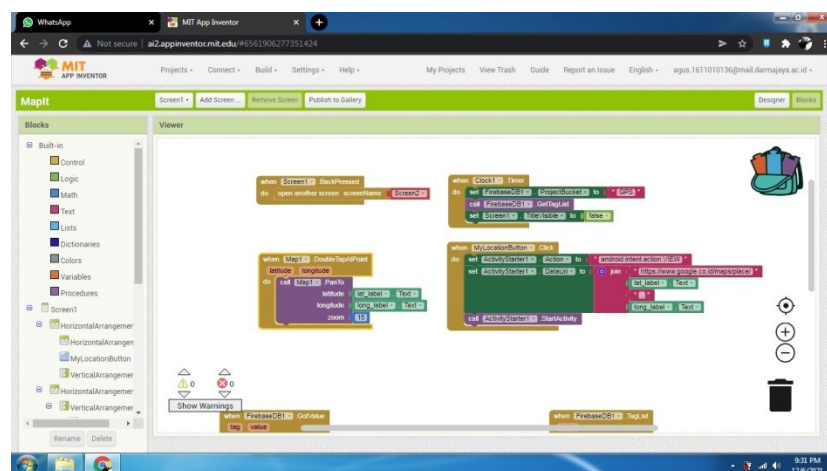
        Firebase.setString("/GPS/Latitude", latitude);
        Firebase.setString("/GPS/Longitude", longitude);
        Firebase.setFloat("/GPS/Kecepatan", kecepatan);
        delay(1000);
    }
}
}

```

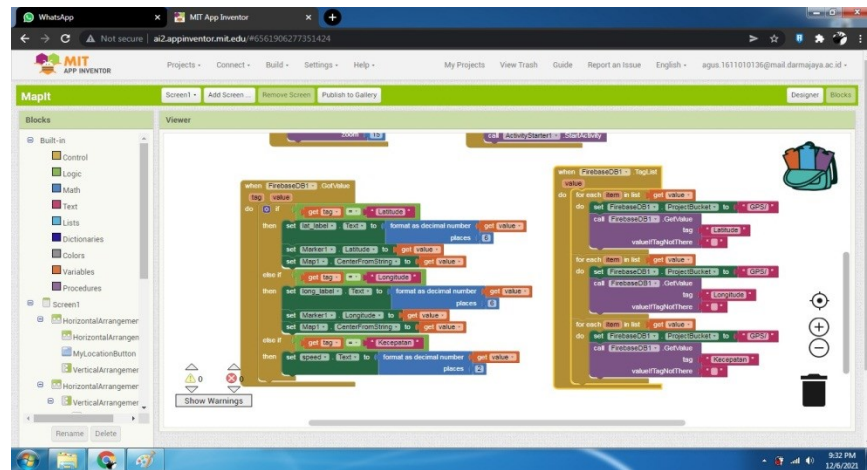
Gambar 3.8 Pemrograman Tracking Lokasi (Arduino)

3.4.3.2 Perancangan Perangkat Lunak (MIT App Inventor)

Pengcodengan sistem menggunakan *MIT App Inventor*. Berikut ini perancangan perangkat lunak atau pengcodengan :



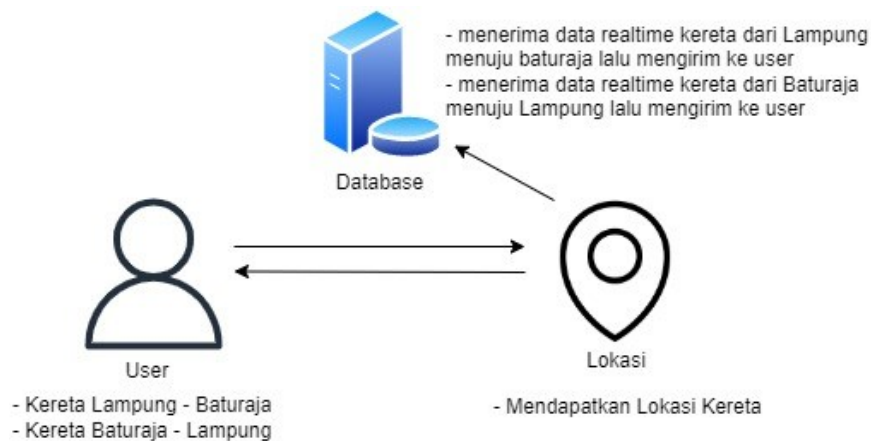
Gambar 3.9 Pemrograman Tracking Lokasi (MIT App Inventor)



Gambar 3.10 Pemrograman Mendapatkan Lokasi (MIT App Inventor)

3.4.4 Arsitektur Sistem

Pada bagian ini, spesifikasi logis diubah kedalam detail teknologi dimana pemrograman dan pengembangan sistem bisa diselesaikan, pada tahap ini aktifitas *coding* dilakukan.



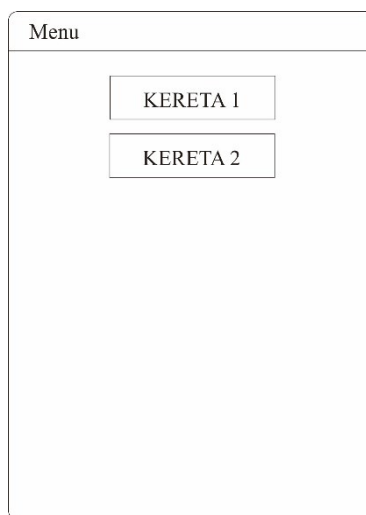
Gambar 3.11 Desain Arsitektur Sistem.

3.4.5 Desain Antarmuka Sistem

Desain program atau aplikasi dilakukan dengan menggunakan perangkat permodelan sistem. Hal ini dilakukan untuk memodelkan sistem aplikasi yang nanti dibangun. Berikut ini desain dari aplikasi yang akan dibangun :

1. Desain Menu Utama

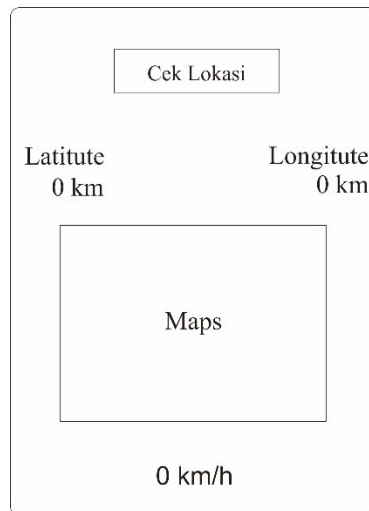
Desain ini menggambarkan menu utama, dimana user bisa melihat lokasi kereta api dan melihat posisi yang ada di *maps*. Berikut ini gambar menu utama.



Gambar 3.12 Desain Menu Utama

2. Desain Lokasi

Desain ini menggambarkan lokasi, di mana user bisa melihat lokasi dari kereta 1 dan kereta 2. Berikut ini desain lokasi ditunjukkan pada gambar 3.12.



Gambar 3.13 Desain Lokasi

3.5 Pengecekan Rangkaian Alat

Pengecekan rangkaian untuk proses pengujian alat dan membuat pembahasan tentang kinerja alat yang telah dibuat. Untuk mendapatkan hasil akhir sesuai yang kita inginkan, maka pengujian alat harus dilakukan dengan teliti dan secara hati-hati. Pengujian yang dilakukan adalah tiap bagian rangkaian alat, agar kita mengetahui unjuk kerja dan fungsi dari tiap-tiap rangkaian.

3.5.1 Pengujian *NodeMCU ESP8266*

Pengecekan pin minimum *nodeMCU ESP8266* dilakukan untuk menguji keluaran pada *port-port*, dengan memberikan nilai *high* pada *port* 0, 1, 2, 3. Berikut tabel pengecekan pin *nodeMCU ESP8266*.

Tabel 3.6 Pengecekan Pin *NodeMCU ESP8266*

No Pin	Tegangan Keluaran(Volt)
AO	0,0
RSV	0,0
RSV	5,01
SD3	0,0
SD2	0,0
SD1	0,0

CMD	0,0
SD0	0,0
CLK	3,97
GND	0,0
3V3	3,02
EN	0,78
RST	0,86
GND	0,0
VIN	0,0
D0	3,02
D1	0,0
D2	3,95
D3	3,95
D4	3,95
3V3	3,94
GND	3,96
D5	3,96
D6	0,0
D7	3,02
D8	0,01
RX	0,01
TX	0,01
GND	0,0
3V3	4,95

Tabel diatas merupakan hasil pengukuran pada pengujian *NodeMCU ESP8266*, pengujian dilakukan dengan agar *NodeMCU ESP8266* bekerja dengan baik atau tidak. Dengan membandingkan tegangan terukur dengan program maupun data sheet.

3.5.2 Pengujian *Module GPS Ublox 6-m*

Pengecekan *module GPS ublox 6-m* dilakukan untuk pengecekan sensor *GPS* dengan mengupload program sebagai berikut:

```
define WIFI_SSID "aselole" // input your home or public wifi name
#define WIFI_PASSWORD "12345678" //password of wifi ssid
```



```

static const int RXPin = 4, TXPin = 5;

static const uint32_t GPSBaud = 9600;

TinyGPSPlus gps;

SoftwareSerial ss(RXPin, TXPin);

String latitude,longitude,kecepatan;

void setup() {

    Serial.begin(115200);

    ss.begin(GPSBaud);

    delay(1000);

    WiFi.begin(WIFI_SSID, WIFI_PASSWORD); //try to connect with wifi

    Serial.print("Connecting to ");

    Serial.print(WIFI_SSID);

    while (WiFi.status() != WL_CONNECTED) {

        Serial.print(".");

        delay(500); }

    Serial.println();

    Serial.print("Connected to ");

    Serial.println(WIFI_SSID);

    Serial.print("IP Address is : ");

    Serial.println(WiFi.localIP()); //print local IP address

    Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); // connect to
    firebase

}

void loop() {

```

```

while (ss.available() > 0)
    if (gps.encode(ss.read())){
float la= (gps.location.lat());
float lo= (gps.location.lng());
float kecepatan = gps.speed.kmph();
        if(kecepatan <=1){
            kecepatan=0; }
latitude = String(la,6);
longitude = String (lo,6);
Serial.print("Location: ");
if (gps.location.isValid() {
    Serial.print(latitude);
    Serial.print (" ");
    Serial.println(longitude);
    Serial.print("Kecepatan: ");
    Serial.println(kecepatan);
    Firebase.setString("/GPS/Latitude", latitude);
    Firebase.setString("/GPS/Longitude", longitude);
    Firebase.setFloat("/GPS/Kecepatan", kecepatan);
    delay(1000);
}
}
}

```