

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

Penelitian dapat diselesaikan secara baik jika dilakukan pengumpulan dan pengolahan data. Pada pengolahan data dilakukan analisis data – data yang diperoleh untuk dapat didiskusikan menuju hasil yang akan disajikan.

4.1 Analisis Kebutuhan

Analisis kebutuhan merupakan tahapan untuk melakukan proses identifikasi, pengumpulan data dan informasi yang digunakan untuk mendukung pembuatan sistem. Agar sistem dapat berjalan secara normal, pada proses analisis kebutuhan dilakukan pengumpulan informasi mengenai perangkat keras dan perangkat lunak. Penelitian ini akan mengembangkan alat serta aplikasi yang menggunakan NodeMCU ESP8266 sebagai pusat kendali. Alat tersebut berfungsi untuk melakukan pelacakan truk distribusi dan cek database stok barang. Untuk aplikasi dikembangkan dengan sistem operasi berbasis *android*.

4.1.1 Analisis Kebutuhan Fungsi

Analisis kebutuhan fungsi merupakan tahapan penetapan fungsi yang terdapat pada sistem untuk menjawab permasalahan yang ada. Fungsi pada sistem dalam penelitian ini sebagai berikut:

- a. Melakukan pelacakan truk distribusi barang.
- b. Mengetahui data stok barang yang dibawa oleh truk distribusi.

4.1.2 Analisis Kebutuhan Masukan

Pada tahapan analisis kebutuhan masukan dilakukan penentuan data yang dibutuhkan. Sistem membutuhkan *input* data berupa *latitude* dan *longitude* yang digunakan untuk menentukan lokasi truk distribusi. Selanjutnya data input barang yang dibawa oleh truk distribusi untuk ditampilkan dalam aplikasi.

4.1.3 Analisis Kebutuhan Keluaran

Penelitian ini dilakukan untuk membuat sistem pelacakan truk distribusi dan cek database stok barang. Berdasarkan fungsi sistem, analisis kebutuhan keluaran dari sistem sebagai berikut :

- a. Mengetahui lokasi truk distribusi.
- b. Mengetahui jenis dan jumlah barang yang dibawa truk distribusi.

4.1.4 Analisis Kebutuhan Perangkat Lunak

Dalam pembuatan sistem dilakukan analisis kebutuhan perangkat lunak. Berdasarkan masalah yang ada, perangkat lunak yang dibutuhkan dalam pembuatan sistem sebagai berikut:

- a. Arduino IDE
- b. Kodular
- c. Autodesk Eagle
- d. Firebase

4.1.5 Analisis Kebutuhan Perangkat Keras

Tahapan analisis kebutuhan perangkat keras dilakukan untuk mengetahui perangkat keras yang dibutuhkan. Untuk menyelesaikan masalah yang ada perangkat keras yang dibutuhkan sebagai berikut:

- a. NodeMCU ESP8266
- b. GPS Ublox 6M
- c. RFID RC522

4.2 Perancangan Sistem

Perancangan sistem yang dimaksud adalah merancang perangkat lunak dan keras yang akan digunakan dalam penelitian.

4.2.1 Perancangan Perangkat Lunak

Aplikasi pelacakan distribusi barang yang akan dilakukan pada penelitian bernama aplikasi *monitoring* dan *tracking* (Montra). Untuk membuat Aplikasi Montra digunakan perangkat lunak kodular. Untuk menghubungkan Aplikasi Montra dengan perangkat keras yang juga dirancang menggunakan perangkat lunak Arduino IDE, untuk membuat perintah ke mikrokontroler NodeMCU ESP8266. Selanjutnya dilakukan penghubungan perangkat keras dengan *firebase*.

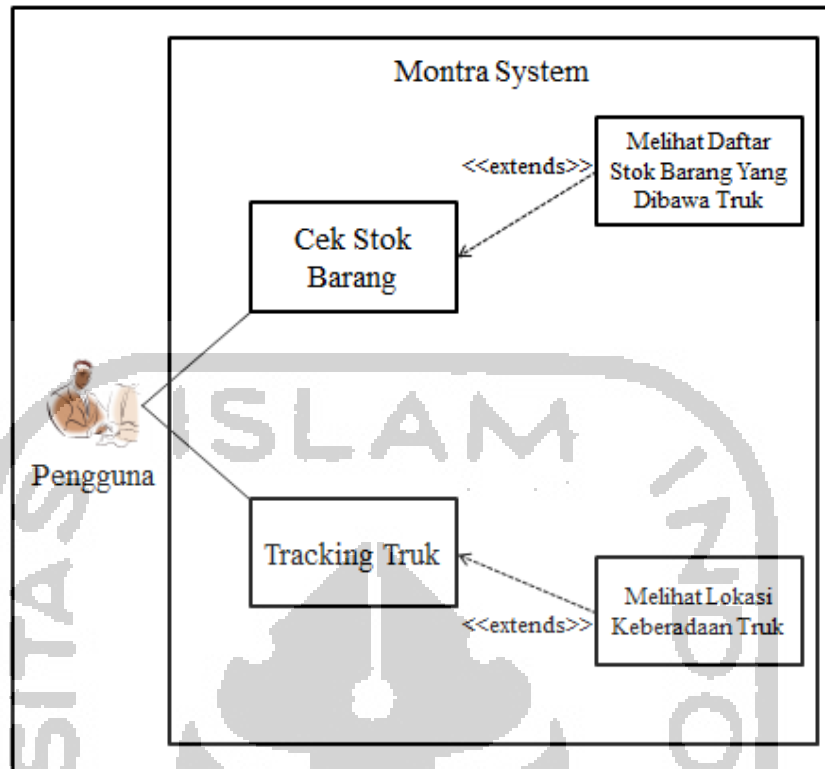
4.2.2 Perancangan Perangkat Keras

Perancangan perangkat keras yang dilakukan merupakan perancangan skema elektrik pada pembuatan *prototype*. Tujuan perancangan skema elektrik untuk menghasilkan

sebuah papan kendali *printed circuit board (PCB)*. PCB dirancang menggunakan perangkat lunak Autodesk eagle. Fungsi dari PCB untuk menghubungkan beberapa modul yang digunakan. Modul tersebut terdiri dari mikrokontroler NodeMCU ESP8266, GPS uBlox neo 6m, RFID RC522. Mikrokontroler NodeMCU ESP8266 berfungsi untuk memberikan perintah ke modul yang terdapat pada PCB.

4.2.3 Use Case Diagram

Use case diagram adalah gambaran grafis yang digunakan untuk menjelaskan fungsi dari model sistem yang akan dikembangkan (Widodo, 2011). Aplikasi Montra akan mengeluarkan diagram yang disebut dengan *use case* diagram montra. Dalam kasus ini terdapat satu aktor sebagai user atau pengguna. Pengguna mempunyai akses untuk menggunakan semua fitur yang terdapat dalam sistem Aplikasi Montra. Seperti Cek Stok Barang dan Pelacakan Truk Distribusi. *Use Case* Diagram Montra pada Gambar 4.1.



Gambar 4.1 Use Case Diagram Montra

Pada gambar 4.1 terdapat satu aktor *user/pengguna* dan dua *case* yaitu cek stock barang dan *tracking* truk, terlihat interaksi antara pengguna dengan *case* yang terdapat dalam sistem.

a. Identifikasi *Use Case* Diagram

Pada sistem Aplikasi Montra terdapat dua *case* yang telah dipetakan berdasarkan kebutuhan fungsi dari sistem. Dibawah ini merupakan keterangan *use case* dari sistem tersebut pada tabel 4.1

Tabel 4. 1 Identifikasi Use Case Diagram

Kode	Nama	Deskripsi
MT01	Cek Stock Barang	Pengguna akan mengetahui barang dan jumlah barang yang sedang dikirim oleh truk distribusi.
MT02	Tracking Truk	Pengguna dapat melihat keberadaan dari truk melalui maps yang ada dalam aplikasi.

b. Skenario *Use Case Diagram*

i. MT01 Cek Stock Barang

Aktor : Pengguna

Prerequisite : -

- a. Sistem akan membaca barang yang keluar dari gudang melewati RFID reader.
- b. Sistem akan mengambil data yang ada dalam database stock barang.
- c. Sistem akan menampilkan data stock barang yang dibawa oleh truk pada aplikasi.

2. MT02 Tracking Truk

Aktor : Pengguna

Prerequisite : -

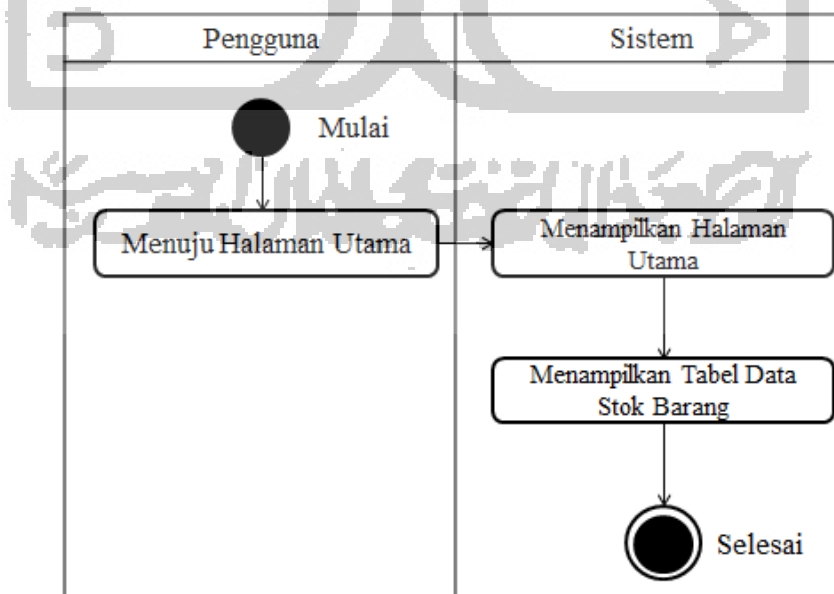
- a. Sistem akan menampilkan halaman splash screen
- b. Sistem memproses data lokasi yang ada dalam database dan merubah latitude dan longitude menjadi titik koordinat.
- c. Sistem menampilkan titik lokasi dari truk distribusi dalam maps yang terdapat di aplikasi.

4.2.4 Activity Diagram

Activity diagram adalah aktivitas dari sebuah sistem atau proses bisnis yang direpresentasikan melalui grafis (Haviluddin, 2011). *Activity* diagram menunjukkan aktivitas dari sistem berupa kumpulan kegiatan, bagaimana dari kegiatan tersebut bekerja, keputusan yang dapat terjadi dari kegiatan tersebut hingga berakhirnya kegiatan. *Activity* diagram menjelaskan tahapan dari aktivitas setiap case yang telah dirancang pada *use case* diagram. Pada *use case* diagram sebelumnya dijelaskan bahwa terdapat dua case yang dapat dilakukan oleh pengguna. Berikut penjelasan dari setiap case menggunakan *activity* diagram :

a. Cek Stock Barang

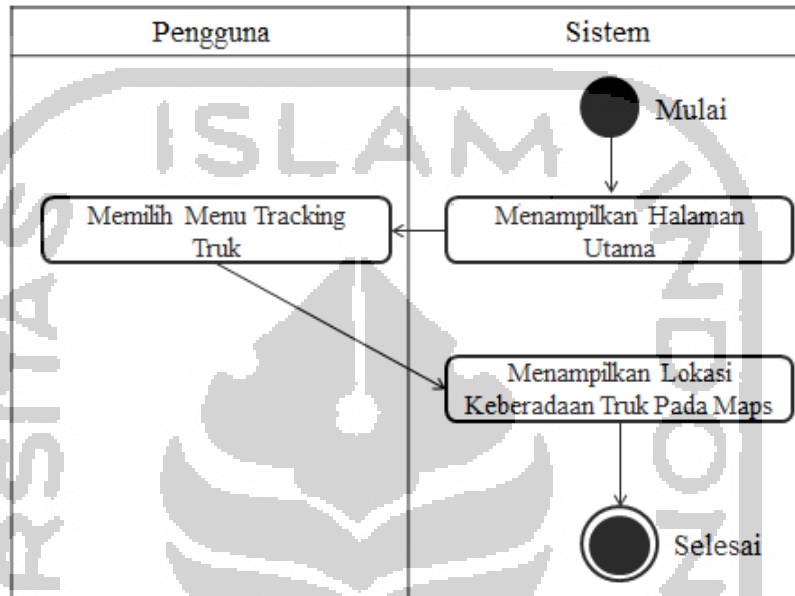
Saat mikrokontroler arduino nano tersambung ke internet, aplikasi akan menampilkan interface awal yang berisi tabel data dari stock barang yang dibawa oleh truk distribusi dan menu tracking truk. Proses melihat tabel data stock barang pada gambar dibawah.



Gambar 4. 2 Activity Diagram Cek Sstok Barang

b. Tracking Truk

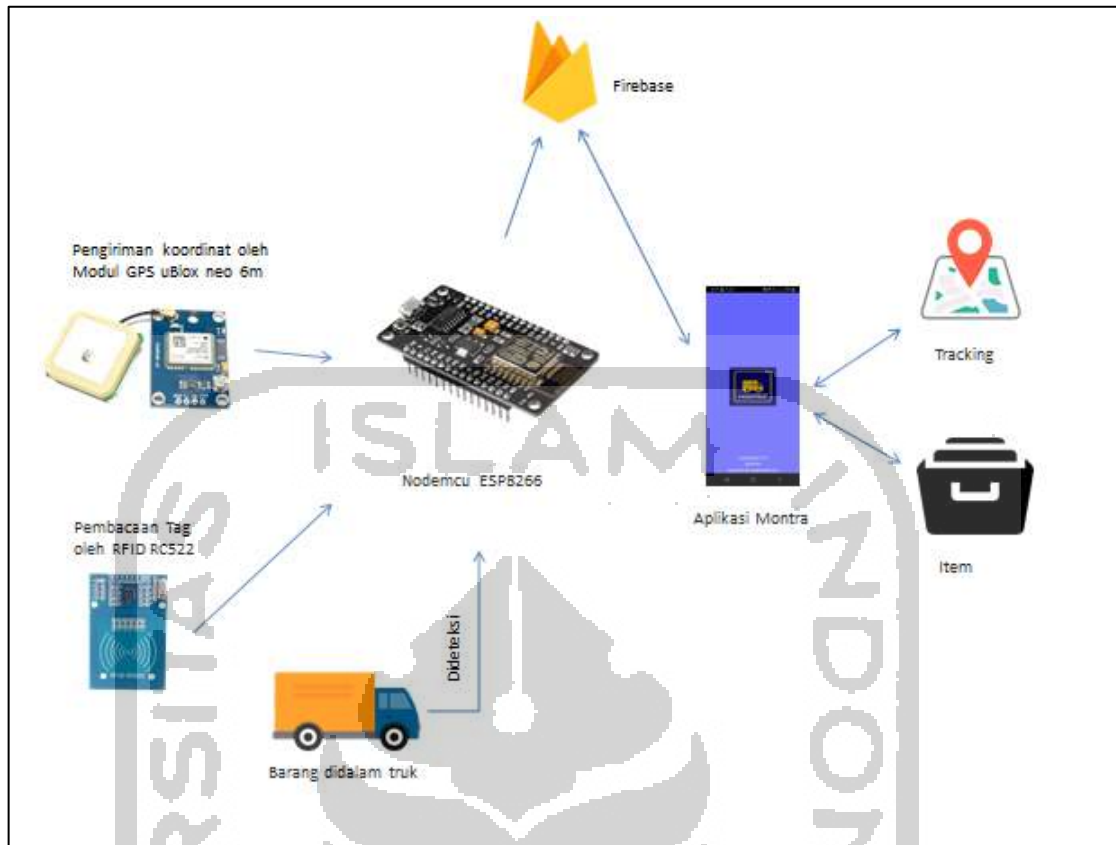
Setelah masuk interface awal, pengguna dapat memilih menu tracking truk untuk pelacakan dari truk distribusi. Proses pelacakan truk dapat dilihat pada gambar dibawah.



Gambar 4. 3 Activity Diagram Tracking Truk

4.2.5 Perancangan Skema Sistem

Perancangan skema sistem dalam penelitian untuk menggabungkan *smartphone* yang menggunakan OS (Operating System) android, mikrokontroler NodeMCU ESP8266, RFID *tag* dan *reader*, dan gps ublox 6m. Perangkat keras tersebut dipasang pada PCB. Penelitian ini bertujuan untuk melakukan tracking truk distribusi dan cek database stok barang. Penjelasan mengenai hubungan setiap perangkat dalam sistem digambarkan pada gambar dibawah.



Gambar 4. 4 Perancangan Skema Sistem

Pada Gambar 4.4 menjelaskan skema dari sistem aplikasi Montra. Pertama, saat barang telah diberi tag RFID keluar dari gudang melewati RFID reader data akan langsung tercatat dalam *database* yang berada di dalam *firebase*. Kedua, pengguna dapat mengecek jenis dan jumlah barang yang dibawa truk distribusi dengan memilih menu *Item* pada aplikasi Montra. Sistem akan mengirim informasi dari *database* ke arduino nano dan diproses untuk ditampilkan dalam aplikasi. Ketika pengguna ingin melakukan pelacakan truk distribusi, maka pengguna dapat memilih menu *track*. Sistem akan merespon dengan mengirimkan titik koordinat dari truk tersebut berupa *latitude* dan *longitude* yang dikirim oleh GPS uBlox 6M ke NodeMCU ESP8266 untuk dikirimkan ke *firebase* agar dapat ditampilkan pada maps yang ada pada aplikasi Montra.

4.2.6 Perancangan Interface

Perancangan interface dilakukan untuk membuat tampilan yang mudah dimengerti dan memiliki desain yang bagus. Hal ini dilakukan agar pengguna nyaman saat menggunakan Aplikasi Montra. Interface dalam aplikasi Montra terbagi menjadi empat interface yaitu :

a. *Greeting Screen*

Halaman *greeting screen* adalah halaman awal dari aplikasi Montra yang akan muncul ketika aplikasi dibuka. Halaman ini hanya menampilkan logo Montra dalam waktu beberapa detik lalu masuk ke halaman utama. Halaman *greeting screen* ini memiliki tampilan seperti pada gambar dibawah.



Gambar 4.5 *Greeting Screen*

b. Home Screen

Pada halaman home screen menampilkan dua menu utama dari aplikasi Montra yaitu Item dan Track. Halaman home screen memiliki tampilan seperti pada gambar dibawah.



Gambar 4. 6 Home Screen

c. Interface Item

Pada halaman Item ini akan ditampilkan tabel data yang memuat jenis dan jumlah barang yang dibawa oleh truk distribusi. Tampilan dari halaman Cek Stok Barang dapat dilihat pada gambar dibawah.



5:56 65%

MONTRA

Dettar ID Card

No. Urut	Item	Status	Temp	Hum	Item	Track
1
2
3
4



Gambar 4.7 Interface Cek Stok Barang

d. Interface Track

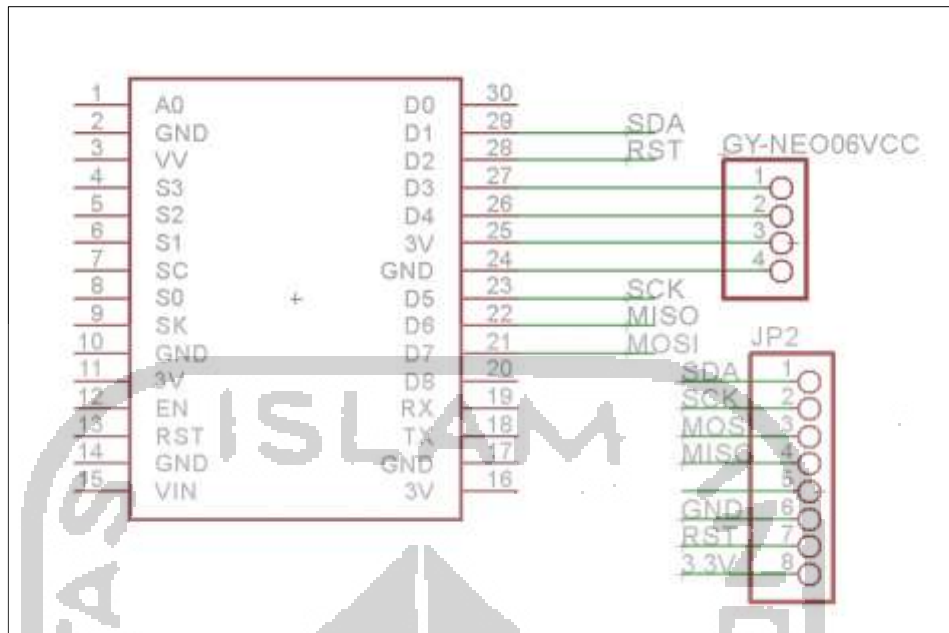
Pada halaman Track pelacakan posisi truk distribusi. Tampilan dari halaman Track terdapat pada gambar dibawah.



Gambar 4. 8 Interface Tracking Truk

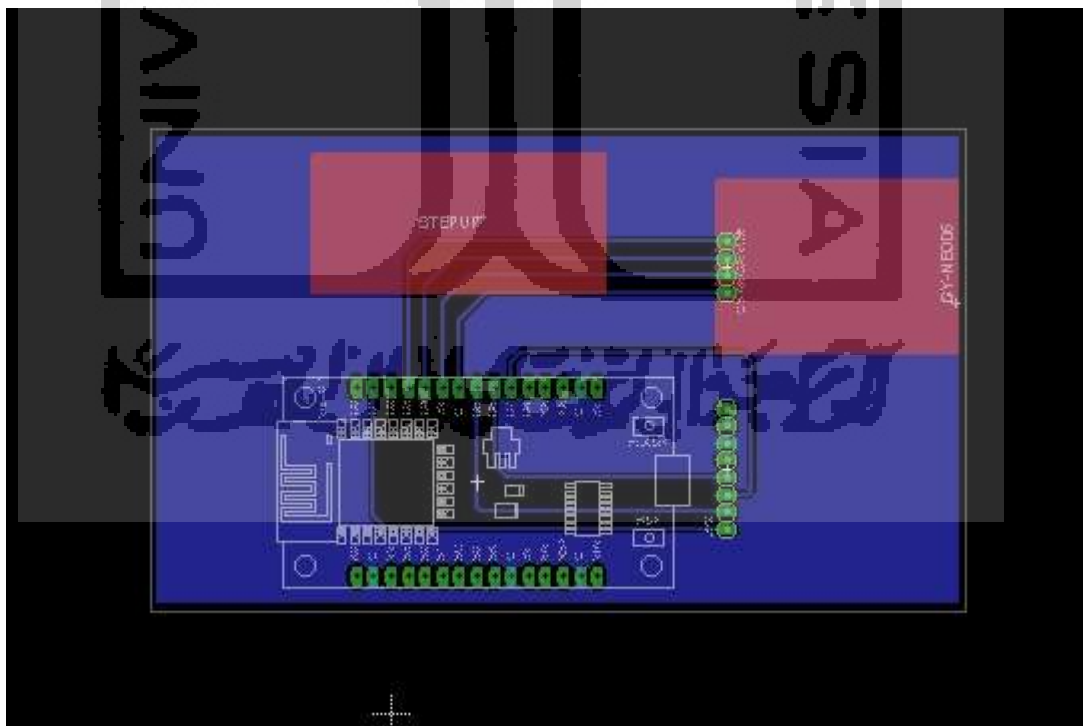
4.2.7 Skematik Rangkaian Elektrik

Skematik rangkaian sistem tracking truk distribusi dan cek stok barang ini menggunakan smartphone android yang terhubung dengan NodeMCU ESP8266 dengan database penyimpanan di *firebase*. Perangkat keras yang digunakan pada sistem ini antara lain GPS uBlox 6M Neo, RFID RC522. Gambar 4.9 menampilkan rangkaian skematik pada sistem.



Gambar 4. 9 Rancangan Skematik

Setelah membuat skematik untuk sistem, selanjutnya pada gambar 4.10 dibuat ke PCB layout untuk di routing sebelum siap untuk di cetak.



Gambar 4. 10 Desain *Layout* PCB

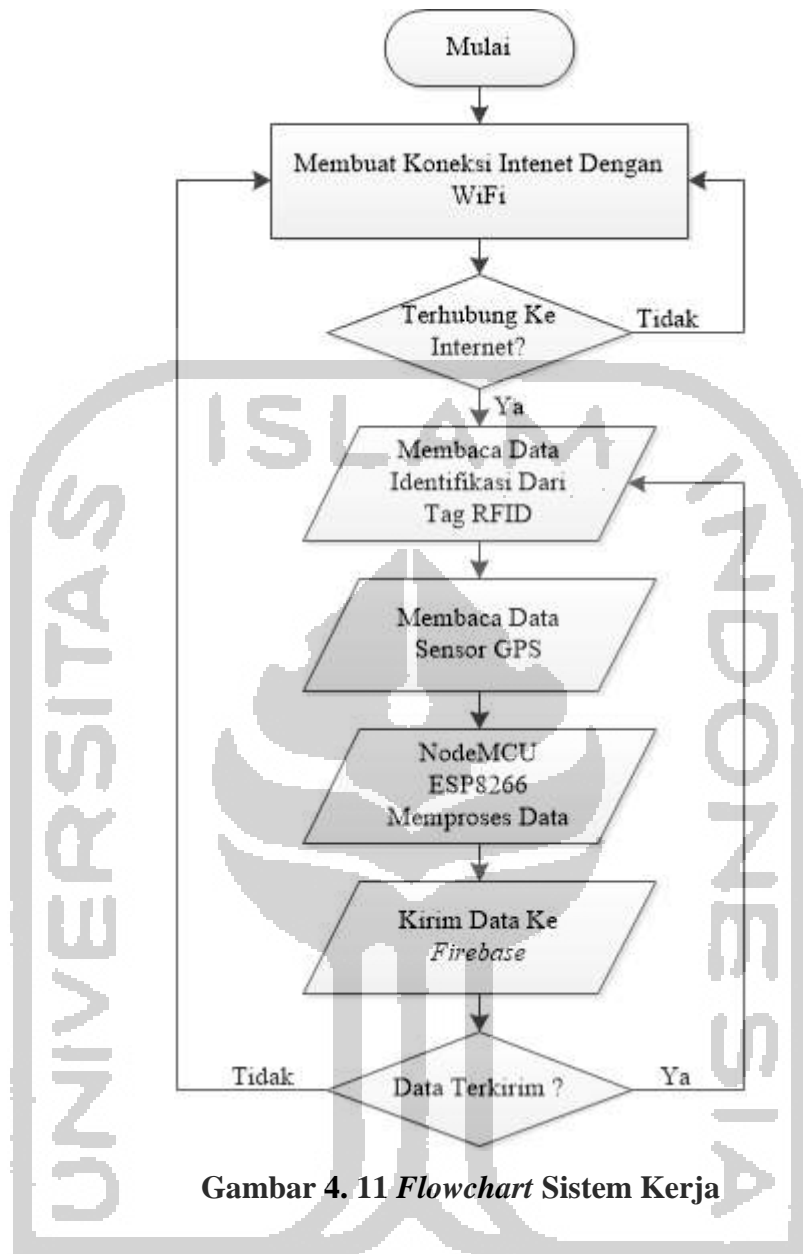
Detail sambungan antara pin pada Arduino Nano dan beberapa modul yang digunakan dijabarkan pada rincian tabel 4.2:

Tabel 4. 2 Rangkaian Pin Sistem

Pin NodeMCU ESP8266	Fungsi	Pin Mode
1	NC	NC
2	NC	NC
3	Data Transfer GPS	Output
4	Data Receiver GPS	Input
5	RST RFID	Output
6	SDA RFID	Input
7	MOSI RFID	Output
8	MISO RFID	Input
9	SCK RFID	Output

4.2.8 Flowchart Sistem Kerja

Cara kerja sistem dan pengiriman data akan dijelaskan dalam *flowchart* sistem kerja. Setelah reader berhasil membaca tag data akan dikirimkan ke *firebase* agar dapat diproses lebih lanjut. Jika sistem akan menampilkan data dari stok barang maka akan mengakses ke *firebase* yang akan ditampilkan di aplikasi. Ketika sistem akan menampilkan lokasi keberadaan truk maka GPS uBlox akan mengirimkan titik koordinat ke NodeMCU ESP8266 agar dapat mengirim data lokasi berupa *latitude* dan *longitude* ke *firebase*. Sistem akan memproses data dari NodeMCU ESP8266 yang sudah ada dalam *firebase* untuk dikirim ke aplikasi server untuk diubah menjadi titik lokasi keberadaan dari truk yang akan ditampilkan di dalam maps. Pada gambar dibawah ini ditunjukkan *flowchart* dari sistem.



Gambar 4. 11 *Flowchart* Sistem Kerja

4.2.9 Perancangan Sistem *Tracking*

Dalam perancangan alat *tracking* truk ini dilakukan dengan bantuan perangkat lunak Arduino IDE yang menggunakan gabungan dari beberapa bahasa pemrograman seperti C++, Java, dan yang lainnya. Pada perancangan ini juga dibantu oleh *library* dari database untuk penyimpanan data lokasi gps berupa *latitude* dan *longitude*. Berikut merupakan baris kode program dari sistem tracking.

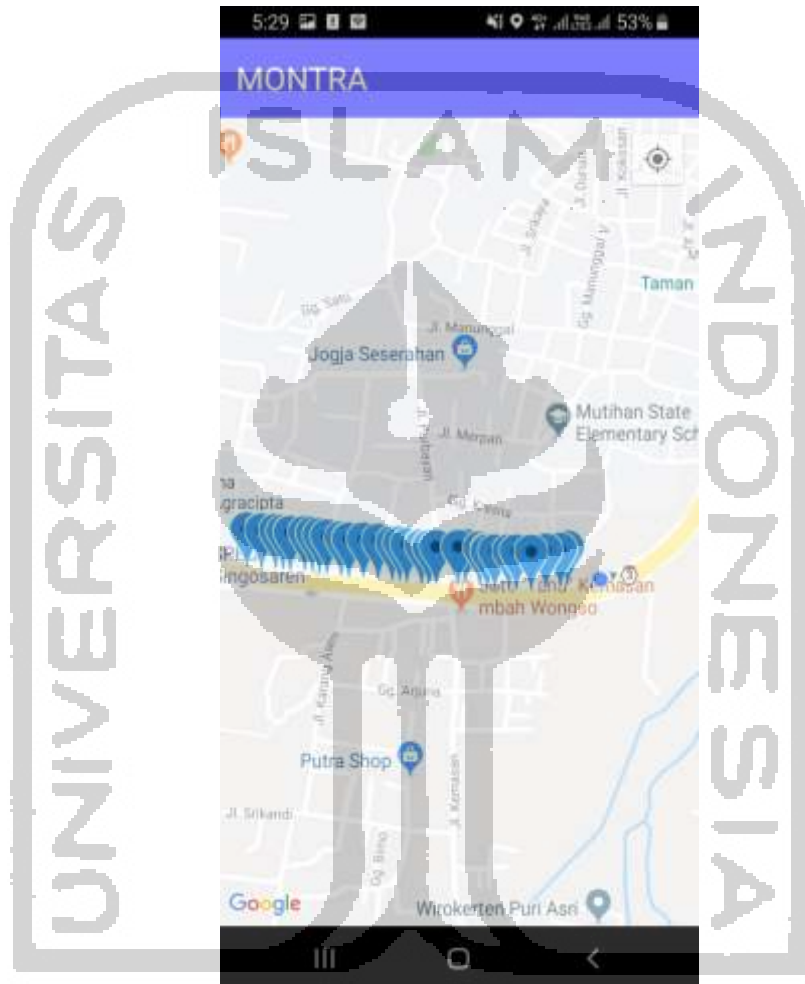
performa sistem dalam mengirim data ke *firebase*. Pengujian dilakukan sebanyak tiga kali, pertama dengan kecepatan 30km/jam, kedua 40km/jam, dan ketiga 60km/jam. Pengujian dilakukan untuk mengetahui keakuratan dari sistem pelacakan. Pelacakan dikatakan akurat apabila interval waktu pengiriman data ke *firebase* tidak signifikan. Berikut merupakan tabel hasil pengujian pengiriman data ke *firebase*.

Tabel 4. 3 Pengiriman Data Kecepatan 30km/jam

Data Ke-	Interval (detik)
1	1
2	1
3	1
4	1
5	1,1
6	1,1
7	1
8	1
9	1
10	1,1
11	1,2
12	1,1
13	1
14	0,9
15	0,9
16	1,1
17	1,1
18	1
19	1,1
20	1
21	1,1
22	1,2
23	1,2
24	1
25	1
26	1
27	1,1
28	1
29	0,9
30	1

Berdasarkan tabel 4.3 didapatkan 30 data hasil waktu pengiriman data dari sistem ke *firebase*. Diketahui interval terkecil dari pengiriman data yaitu 0,9 detik dan

terbesar 1,2 detik. Rata – rata nilai interval waktu pengiriman data pada kecepatan 30km/jam adalah 1.04 detik. Proses pelacakan pada kecepatan 30km/jam dikatakan akurat karena waktu interval pengiriman data yang tidak signifikan.



Gambar 4. 14 Hasil Pelacakan pada Kecepatan 30km/jam

Pada gambar 4.14 dapat dilihat hasil pelacakan yang berjalan dengan baik pada kecepatan 30km/jam. Hal ini disebabkan oleh kecepatan kendaraan yang tidak terlalu kencang yang membuat tanda koordinat lokasi terlihat berdekatan antara satu dengan yang lainnya. Hanya terdapat jarak yang sedikit berbeda saat pertengahan proses pelacakan. Hal ini terjadi karena koneksi internet yang kurang stabil.

Tabel 4. 4 Pengiriman Data Kecepatan 40km/jam

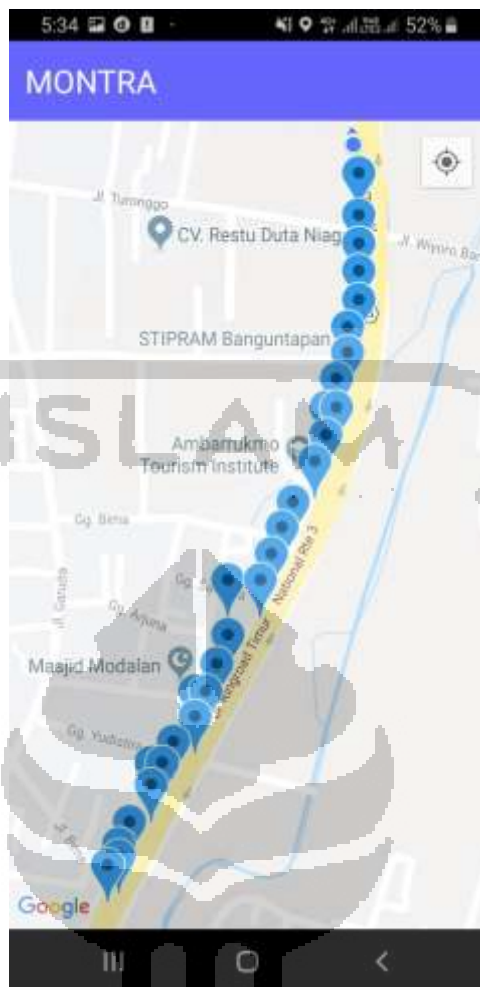
Data Ke-	Interval (detik)
1	1
2	1
3	1,2
4	1
5	1,1
6	1,2
7	1
8	1
9	0,9
10	1,2
11	1,1
12	1
13	1
14	1
15	1,3
16	1,1
17	1,2
18	1,1
19	1
20	1
21	1,2
22	1,1
23	1,2
24	1
25	1
26	1,2
27	1,1
28	1
29	1,2
30	1,1

Berdasarkan tabel 4.4 interval terkecil dari pengiriman data yaitu 0,9 detik dan terbesar 1,3 detik. Rata – rata nilai interval waktu pengiriman data pada kecepatan 40km/jam adalah 1.08 detik. Pengujian proses pelacakan pada kecepatan 40km/jam mendapatkan nilai rata – rata interval waktu yang sedikit lebih besar dibandingkan dengan kecepatan 30km/jam.

Tabel 4. 5 Pengiriman Data Kecepatan 60km/jam

Data Ke-	Interval (detik)
1	1,3
2	1
3	1,2
4	1
5	1,1
6	1,2
7	1
8	1
9	1
10	1,2
11	1,1
12	1
13	1,2
14	1
15	1,2
16	1,1
17	1,2
18	1,1
19	1
20	1
21	1,2
22	1,1
23	1,2
24	1,3
25	1
26	1,2
27	1,1
28	1
29	1
30	1

Berdasarkan tabel 4.5 interval terkecil dari pengiriman data yaitu 1 detik dan terbesar 1,3 detik. Rata – rata nilai interval waktu pengiriman data pada kecepatan 40km/jam adalah 1.1 detik. Pengujian proses pelacakan pada kecepatan 60km/jam rata – rata interval waktu pengiriman data lebih besar dibandingkan dengan kecepatan 30km/jam dan 40km/jam.



Gambar 4. 16 Hasil Pelacakan pada Kecepatan 60km/jam

Hasil pelacakan pada gambar 4.16 memperlihatkan tanda koordinat lokasi jaraknya tidak terlalu dekat. Hal ini dapat disebabkan oleh kecepatan dan koneksi internet yang kurang stabil saat pengujian. Dapat disimpulkan bahwa pelacakan yang akurat bergantung pada kecepatan dan koneksi internet yang stabil.

4.3.2 Pengujian Aplikasi Montra

Pengujian Aplikasi Montra dilakukan menggunakan metode *black-box testing*. Pengujian menggunakan metode ini untuk mengetahui apakah fungsi – fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang

dibutuhkan. Pengujian ditekankan pada fungsional dari aplikasi montra. Tahap ini berisi serangkaian pengujian fungsi dan tombol dari aplikasi. Berikut merupakan hasil dari pengujian aplikasi dengan metode *black-box testing*.

Tabel 4. 6 Hasil Pengujian Aplikasi dengan *Black-Box Testing*

Fungsi Yang Diuji	Kondisi Awal	Kondisi Akhir Yang Diharapkan	Hasil Pengujian
Pengujian Aplikasi	Pengguna belum menjalankan aplikasi	Masuk ke Aplikasi dan Menampilkan halaman greeting	Berhasil
Pengujian Halaman Utama	Pengguna berada pada halaman utama aplikasi	Menampilkan menu yaitu Track untuk melakukan pelacakan dan Item untuk mengecek stok barang	Berhasil
Pengujian Menu Track	Pengguna berada pada halaman utama aplikasi	Sistem menampilkan lokasi dari truk dan titik koordinat akan mengikuti kemana arah yang dituju oleh truk	Berhasil
Pengujian Menu Item	Pengguna berada pada halaman utama aplikasi	Menampilkan login agar dapat melihat stok barang	Berhasil

Fungsi Yang Diuji	Kondisi Awal	Kondisi Akhir Yang Diharapkan	Hasil Pengujian
Pengujian Login	Pegguna berada pada halaman menu Item	Menampilkan data berupa tabel nomor id barang, waktu ketika proses pembacaan pada reader RFID, nama barang, jumlah barang terdistribusi, jumlah total barang saat pengiriman, dan jumlah sisa barang	Berhasil
Pengujian Tombol Tambah dan Hapus Data	Pegguna berada pada halaman menu Item	Menampilkan data barang yang telah diubah	Berhasil
Pengujian Tombol Keluar	Pegguna berada pada halaman utama	Keluar dari aplikasi	Berhasil