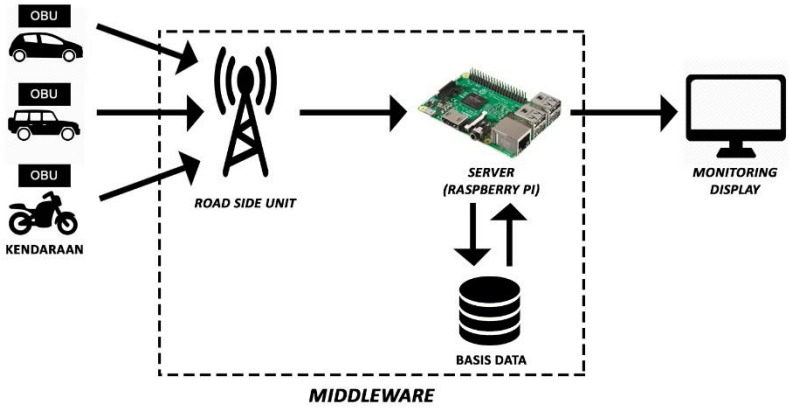
# BAB III
# PERANCANGAN SISTEM

## 3.1 Gambaran Umum Sistem

Sistem *middleware* ini terdiri dari beberapa komponen, setiap komponen saling terhubung dan bertukar informasi. Berikut adalah gambaran umum sistem ditunjukkan pada gambar 3.1.




Gambar 3.1 Gambaran Umum Sistem *Middleware*

Dari Gambar 3.1 dijelaskan bahwa pada sistem *middleware* ini akan menerima hasil pengiriman data (akuisisi data) dari beberapa kendaraan melalui perangkat *Road Side Unit* (RSU) dalam bentuk format data *string*, kemudian di *import* kedalam format Excel yang ada pada *Raspberry Pi*. Nantinya dari format Excel akan diubah kedalam format data SQL untuk disimpan ke dalam basis data. Proses penyimpanan data ke dalam basis data melalui *server* yang menggunakan *Raspberry Pi* juga. Kemudian setelah data yang diterima tersimpan pada basis data, jika ada perangkat lain yang ingin mengakses data kendaraan, maka *server* akan mengirimkan data tersebut. Pada sistem *middleware* ini berfungsi untuk menghubungkan komponen lain bersama, misalnya aplikasi *enterprise* serta mengelola data diseluruh sistem, dengan ini memungkinkan dapat menyebarkan format data ke sistem lain.

Berikut adalah keterangan dari gambar 3.1:

1. *OnBoard Unit* (OBU) berfungsi untuk mengirimkan informasi berupa data dari kendaraan menuju *Road Side Unit* (RSU).

2. *Road Side Unit* (RSU) berfungsi untuk menerima data dari *OnBoard Unit* (OBU).
3. *Server* (*Raspberry Pi*) berfungsi sebagai media pemroses dan penyimpanan data.
4. *Monitoring Display* berfungsi untuk menampilkan data yang telah diolah oleh *server* guna melakukan pemantauan kendaraan.

## 3.2 Komponen Sistem *Middleware*

Untuk perancangan *middelware* sistem pemantauan kendaraan dibutuhkan beberapa perangkat lunak dan keras pada *server* dan basis data yang digunakan, yaitu sebagai berikut:

1. Satu buah Mini-PC *Raspberry Pi* 3 Model B dan memori SD *Card* kapasitas 16 GB.
2. Sistem operasi Raspbian Jessie.
3. Basis data MySQL.
4. *Python* 3.4.
5. *Text editor*.

## 3.3 Proses Sistem *Middleware*

Sistem ini akan menerima data kendaraan yang dikirimkan oleh kendaraan melalui perangkat *Road Side Unit* dan menyediakan format data untuk perangkat *monitoring display* yang ingin mengakses dan mengolah data kendaraan yang dikirimkan, berikut adalah langkah-langkahnya:

1. Kendaraan diberikan tanda pengenal agar bisa teridentifikasi pada basis data dan sistem. Pada sistem ini kendaraan diberikan tanda pengenal yaitu berupa "id".
2. Kendaraan berkomunikasi dengan perangkat *Road Side Unit* dan mengirimkan data terhadap perangkat *Road Side Unit* tersebut.
3. Perangkat *Road Side Unit* mengakses *server* melalui *serial communication* dengan *port* yang telah disediakan.
4. Perangkat *Road Side Unit* mengirimkan beberapa *query* pada *port* dengan format yang telah ditentukan.

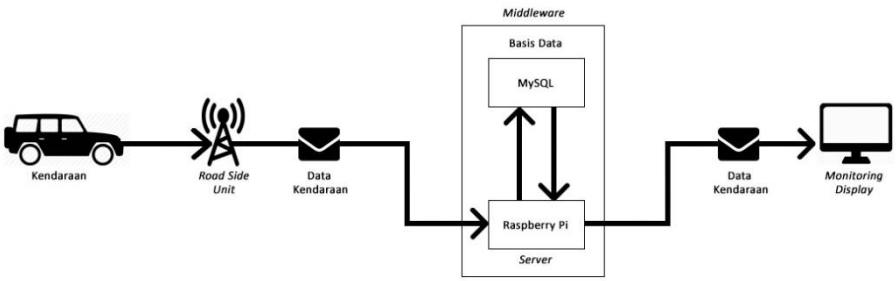
5. *Server* akan memisahkan data kendaraan yang dikirimkan berdasarkan tanda pengenal kendaraan ke dalam basis data.
6. Bila basis data berhasil menerima data dari *server*, basis data akan memberikan respon terhadap *server*.
7. Data kendaraan yang telah berhasil disimpan ke dalam basis data diolah oleh *server* ke dalam format basis data SQL.
8. Data kendaraan yang memiliki format SQL dapat dibaca oleh *monitoring display* yang ingin mengolah data kendaraan melalui *server* dengan protokol HTTP dan dengan halaman yang telah disediakan.

## 3.4 Konsep Pengiriman Data *Middleware*

*Middleware* ini bertujuan agar perangkat *monitoring display* dapat berkomunikasi, sistem ini yang menjadi jembatan antara kendaraan dan perangkat *monitoring display*. Dengan adanya sistem ini, data kendaraan yang dikirimkan oleh setiap kendaraan mudah untuk diolah dan dimanfaatkan oleh perangkat *monitoring display*. Data kendaraan yang akan diterima oleh sistem ini untuk perangkat *monitoring display* adalah sebagai berikut:

1. ID kendaraan.
2. Plat nomor kendaraan.
3. Koordinat lokasi kendaraan berdasarkan *latitude* dan *longitude*.
4. Waktu dan tanggal pengiriman data.

Berikut adalah konsep pengiriman data *middleware* ditunjukkan pada gambar 3.2.




Gambar 3.2 Konsep Pengiriman Data *Middleware*

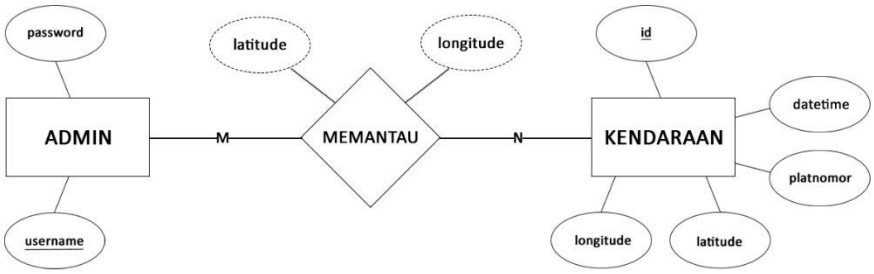
### 3.4.1 Konfigurasi *Road Side Unit* Sebagai Penerima Data

Perangkat *Road Side Unit* (RSU) berfungsi sebagai penerima (*receiver*) data yang ditransmisikan oleh kendaraan melalui *Onboard Unit* (OBU). Memanfaatkan

Mini PC Raspberry Pi, berikut *source code* pada Mini PC Raspberry Pi yang memfungsikan perangkat *Road Side Unit* sebagai penerima data.

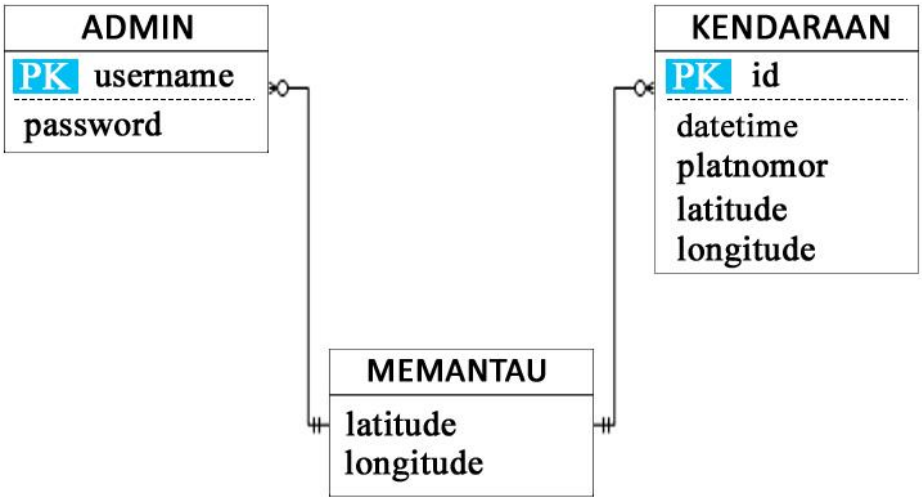
### 3.4.2 Konfigurasi Basis Data

Pada pemodelan basis data diperlukan *Entity Relationship Diagram (ERD)* menggambarkan hubungan antar data yang akan disimpan, diambil dan dipanggil kembali untuk keperluan tertentu. Gambar 3.3 dibawah ini merupakan diagram hubungan entitas dan atribut pada *middleware* untuk sistem pemantauan kendaraan.




Gambar 3.3 *Entity Relationship Diagram*

Untuk keterangan mengenai entitas dan atribut pada gambar 3.3 dapat dilihat dari struktur tabel ERD pada gambar 3.4 berikut.




Gambar 3.4 Struktur Tabel ERD

Tipe data *field* pada tabel adalah sebagai berikut:

a. Tabel Admin

Tabel 3.1 Tipe Data *Field* Tabel Admin

| No. | Nama Kolom | Tipe Data | Keterangan |
|---|---|---|---|
| 1 | username | varchar(20) | PRIMARY KEY |
| 2 | password | varchar(20) | |

b. Tabel Kendaraan

Tabel 3.2 Tipe Data *Field* Tabel Kendaraan

| No. | Nama Kolom | Tipe Data | Keterangan |
|---|---|---|---|
| 1 | id_kendaraan | integer(11) | PRIMARY KEY |
| 2 | datetime | CURRENT_TIMESTAMP() | |
| 3 | platnomor | varchar(50) | |
| 4 | latitude | varchar(50) | FOREIGN KEY (tb_kendaraan.latitude) |
| 5 | longitude | varchar(50) | FOREIGN KEY (tb_kendaraan.longitude) |

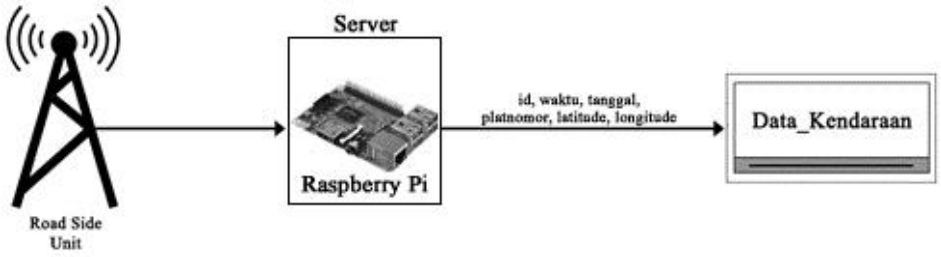
c. Tabel Memantau

Tabel 3.3 Tipe Data *Field* Tabel Memantau

| No. | Nama Kolom | Tipe Data | Keterangan |
|---|---|---|---|
| 1 | latitude | varchar(50) | FOREIGN KEY (tb_kendaraan.latitude) |
| 2 | longitude | varchar(50) | FOREIGN KEY (tb_kendaraan.longitude) |

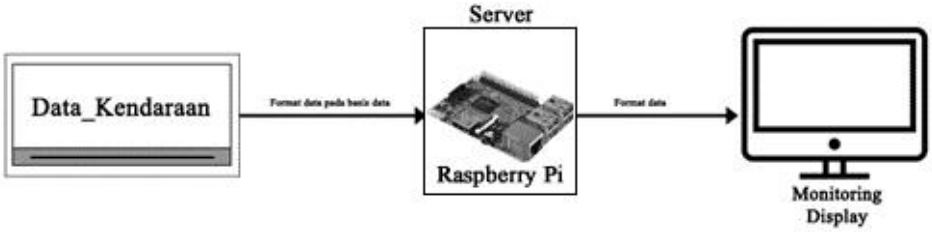
### 3.4.3 Konfigurasi Mini PC Raspberry Pi Sebagai *Server*

Selanjutnya setelah selesai konfigurasi penerima data, maka selanjutnya dapat dilakukan konfigurasi pada Mini PC Raspberry Pi sebagai *server*. Konfigurasi yang harus dilakukan adalah sebagai berikut:




Gambar 3.5 Konsep Pencocokkan Data Ke Basis Data

Pada gambar 3.5 ditunjukkan bahwa *Road Side Unit* (RSU) akan meneruskan data kendaraan menuju ke *server* (Mini PC Raspberry Pi), kemudian *server* akan mencocokkan data kendaraan yang diterima ke dalam basis data.




Gambar 3.6 Konsep Setelah Pencocokkan Data Dengan Basis Data

Pada gambar 3.6 ditunjukkan bahwa *server* akan mengirimkan data kendaraan menuju ke *monitoring display* sesuai dengan format yang dibutuhkan setelah data kendaraan cocok dengan data di dalam basis data.

### 3.4.4 Konfigurasi Bahasa Pemrograman *Python*

Selanjutnya melakukan konfigurasi pada *python* agar *server* dapat berfungsi sesuai yang diharapkan, dimana beberapa kendaraan dapat terpenuhi dalam penyimpanan data.

#### 3.4.4.1 Konfigurasi Koneksi *Python* Dengan Basis Data

Konfigurasi ini dilakukan agar bahasa pemrograman *python* dapat terhubung dengan basis data SQL. Data yang akan diterima dan diolah oleh *server* (Mini PC Raspberry Pi) akan disimpan ke basis data menggunakan bahasa pemrograman *python* sebagai koneksi antara data dengan basis data.

### 3.4.4.2 Konfigurasi *Python* Untuk Pencocokkan Data

Konfigurasi ini dilakukan agar kendaraan dapat terkoneksi dengan *server* melalui *Road Side Unit* untuk dapat melakukan pencocokan data kendaraan yang sesuai pada basis data. Pada kendaraan yang terkoneksi akan mengirimkan data "id kendaraan", data tersebut harus memiliki identitas berupa "id" pada setiap pengiriman. Data yang dikirimkan akan dipisahkan berdasarkan dengan basis datanya. Perangkat *Road Side Unit* mengirimkan data melalui komunikasi *serial* serta *query* untuk data antara lain :

1. "id" untuk identitas/nomor setiap kendaraan.
2. "platnomor" untuk data plat nomor setiap kendaraan.
3. "lat" untuk data lokasi koordinat *latitude.*
4. "lon" untuk data lokasi koordinat *longitude*.
5. "datetime" untuk data tanggal dan waktu penerimaan data kendaraan.

Konfigurasi tersebut berfungsi sebagai pengurai data kendaraan dan menempatkannya pada basis data yang telah ditentukan.

### 3.4.4.3 Konfigurasi Mengganti Format Data Ke Tabel HTML

Pada konfigurasi ini dilakukan supaya perangkat lain berupa *monitoring display* dapat melihat dan mengakses data melalui *web browser*. Ini bertujuan supaya data kendaraan yang mempunyai format tabel HTML bisa diakses oleh protokol HTTP. Sebelumnya, perlu melakukan koneksi antara PHP dan basis data terlebih dahulu.

```
// Koneksi PHP dengan Basis Data
<?php
// isi nama host, username mysql, dan password mysql anda
$host = mysql_connect("localhost","root","rudysetiawan");

if($host){
        echo "Host Connection Success<br/>";
}else{
        echo "Connection Failed<br/>";
}
```

```
// isikan dengan nama database yang akan di hubungkan
$db = mysql_select_db("dataku");

if($db){
        echo "Database Connection Success";
}else{
        echo "Database Connection Failed";
}
?>
```

Selanjutnya, merubah format data SQL menjadi format HTML. Berikut adalah *source code* untuk merubah format data SQL menjadi format HTML.

```
<?php
$link=mysql_connect('localhost','root','rudysetiawan');

if (!$link)
{
  die("MySQL connection failed");
}

$result=mysql_query('USE dataku');
if (!$result)
{
  die("Database fail to used");
}
$result=mysql_query('SELECT * FROM kendaraan');
?>
<!DOCTYPE html>
<head>
  <meta charset="UTF-8">
  <title>Pemantauan Transportasi Darat</title>
  <style>
```

```
  h3{
    text-align:center; }
  table {
    border-collapse:collapse;
    border-spacing:0;
    font-family:Arial, sans-serif;
    font-size:16px;
    padding-left:300px;
    margin:auto; }
  table th {
    font-weight:bold;
    padding:10px;
    color:#fff;
    background-color:#2A72BA;
    border-top:1px black solid;
    border-bottom:1px black solid;}
  table td {
    padding:10px;
    border-top:1px black solid;
    border-bottom:1px black solid;
    text-align:center; }
 tr:nth-child(even) {
   background-color: #DFEBF8; }
  </style>
</head>
<body>
  <h3>Data Pemantauan Transportasi Darat</h3>
<table>
<tr>
  <th>ID</th>
  <th>Plat Nomor</th>
  <th>Latitude</th>
```

```
  <th>Longitude</th>
  <th>Tanggal & Waktu</th>
</tr>
<?php
while ($row=mysql_fetch_array($result))
{
  echo "<tr>";
  echo "<td>".$row['id']."</td>";
  echo "<td>".$row['platnomor']."</td>";
  echo "<td>".$row['lat']."</td>";
  echo "<td>".$row['lon']."</td>";
  echo "<td>".$row['datetime']."</td>";
  echo "</tr>";
}
?>
</table>
</body>
</html>
```

### 3.4.5 Antarmuka Komunikasi

Sistem antarmuka data validasi menggunakan antarmuka komunikasi *client-server* yang terhubung melalui jaringan lokal.

## 3.5 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional dibagi menjadi dua, yaitu analisis perangkat keras dan analisis perangkat lunak. Analisis perangkat keras bertujuan untuk memudahkan proses perancangan dan implementasi *middleware* untuk sistem pemantauan kendaraan. Sedangkan analisis perangkat lunak bertujuan untuk mendukung kebutuhan perangkat keras dalam proses perancangan dan implementasi *middleware* untuk sistem pemantauan kendaraan.

### 3.5.1 Analisis Kebutuhan Perangkat Keras

Kebutuhan minimum perangkat keras yang dibutuhkan untuk menjalankan *middleware* dapat dilihat pada Tabel 3.4 di bawah.

Tabel 3.4 Kebutuhan Perangkat Keras

| No | Perangkat Keras | Spesifikasi |
|---|---|---|
| 1. | 1 buah Mini PC Raspberry Pi 3 Model B | • Audio outputs: 3.5mm jack<br>• CPU: 700 MHz ARM1176JZF-S core (ARM11 family)<br>• GPU: Broadcom VideoCore IV, OpenGL ES 2.0, 1080p30 h.264/MPEG-4 AVC high-profile decoder<br>• Memory (SDRAM): 512 Megabytes (MB)<br>• Onboard storage: SD, MMC, SDIO card slot<br>• SoC: Broadcom BCM2835 (CPU, GPU, DSP, and SDRAM)<br>• Storage via SD/ MMC/ SDIO card slot<br>• Video outputs: Composite RCA, HDMI (High-Definition Multimedia Interface)<br>• 10/100 Ethernet RJ45 onboard network |
| 2. | 1 buah SD *Card* | • 16 GB capacity<br>• Class 10 (48Mbps data transfer rate) |

### 3.5.2 Analisis Kebutuhan Perangkat Lunak

*Middleware* untuk sistem pemantauan kendaraan ini membutuhkan perangkat lunak pendukung agar dapat berjalan pada perangkat keras yang sesuai dengan spesifikasinya pada Tabel 3.4. Berikut pada Tabel 3.5 menunjukkan kebutuhan perangkat lunak.

Tabel 3.5 Kebutuhan Perangkat Lunak

| No. | Perangkat Lunak | Nama Perangkat Lunak |
|---|---|---|
| 1. | Sistem Operasi Raspberry Pi | Raspbian Jessie |
| 2. | Bahasa Pemrograman | Python |

### 3.6 Analisis Kebutuhan Fungsional

Dalam merancang sistem ini menggunakan Diagram Hubungan Entitas. Pada dasarnya sistem ini melakukan validasi data, mengakuisisi data, mengolah data, dan mengeluarkan respon hasil pengolahan data.