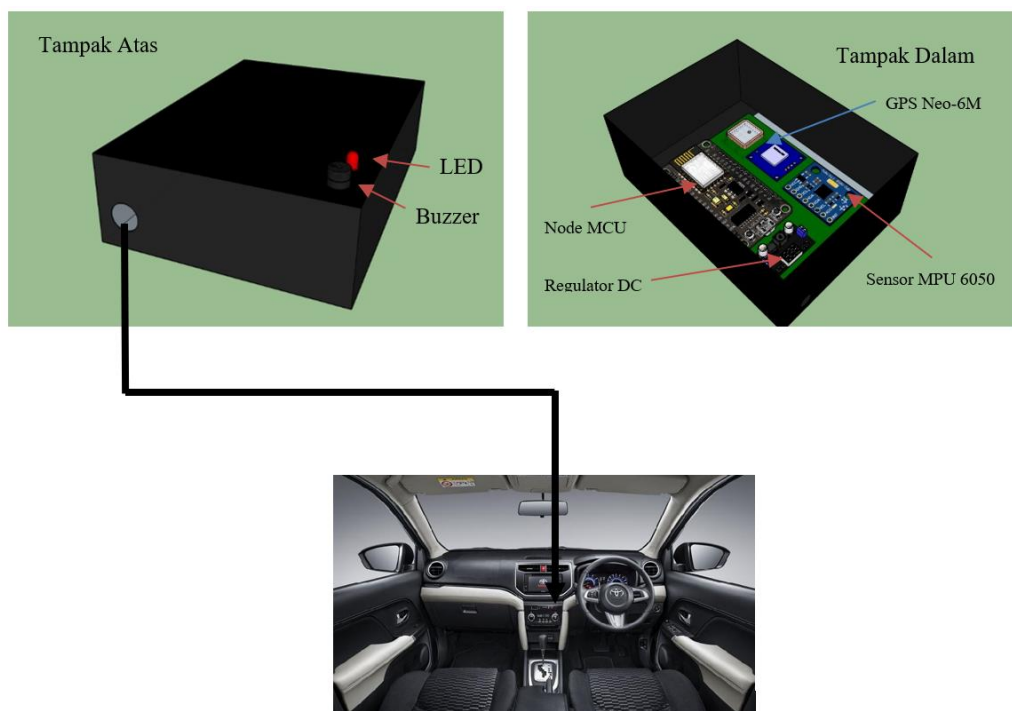


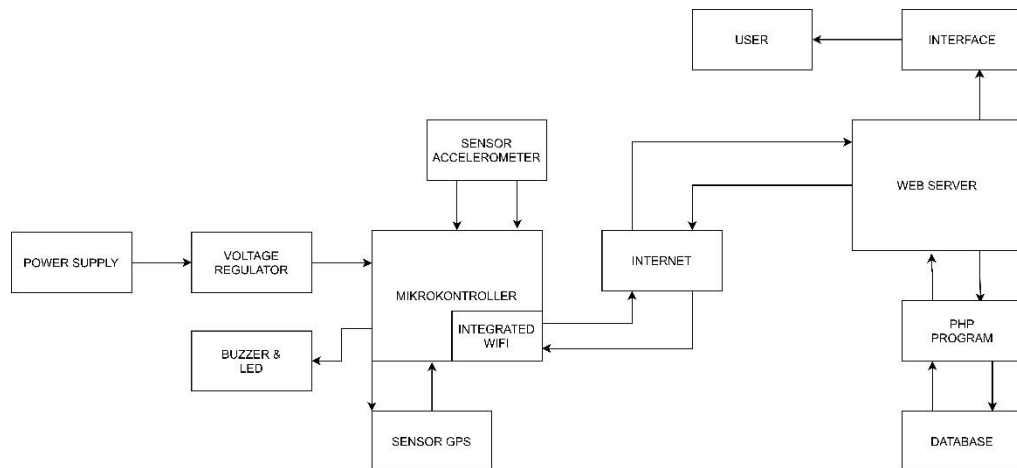
BAB III PERANCANGAN

3.1 Diagram Blok Dari PEDEBATSY

Pothole Detection Based On IoT For Supporting Smar City's System (PEDEBATSY) merupakan suatu sistem yang disusun oleh 3 subsistem inti, yaitu subsistem mikrokontroller, subsistem *website* serta subsistem aplikasi *mobile*. Pada subsistem mikrokontroller merupakan perangkat keras (*hardware*), sedangkan pada subsistem *website* dan aplikasi *mobile* merupakan perangkat lunak (*software*). Gambar 3 merupakan blok diagram dari sistem PEDEBATSY.



Gambar 3. Desain 3D Dari Sistem.

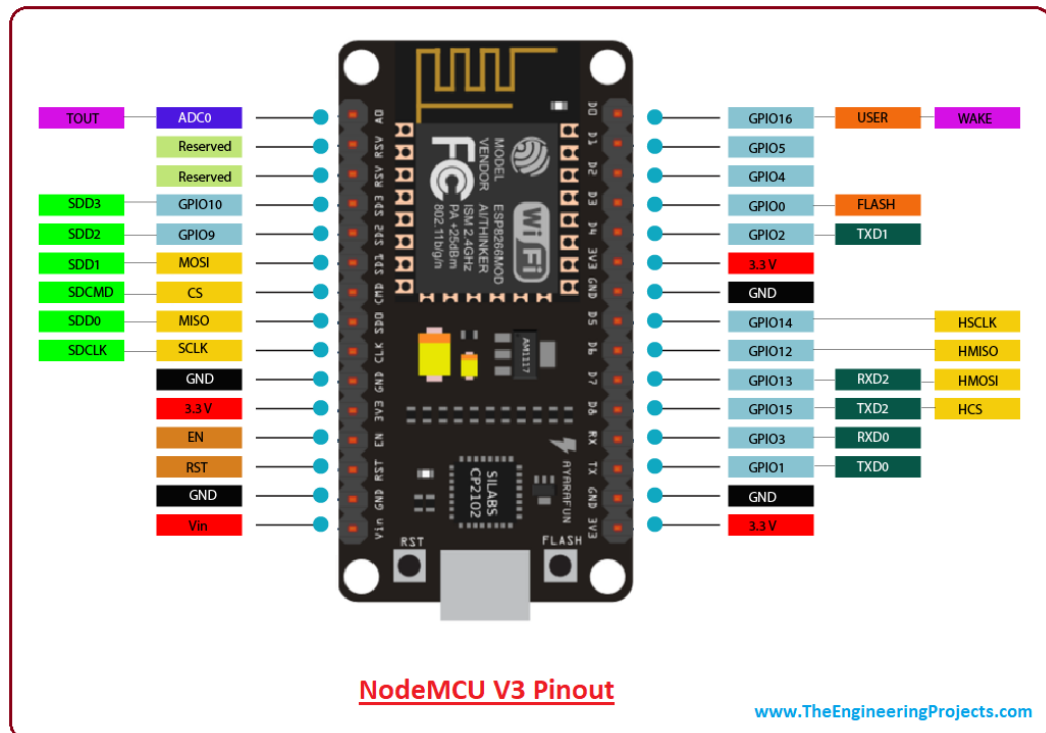


Gambar 4. Blok Diagram Sistem.

Perhatikan Gambar 4. Secara umum cara kerja dari sistem PEDEBATSY adalah dengan melakukan pendeteksian awal letak lubang pada subsistem mikrokontroller yang dimana dengan bantuan modul *global positioning system* (GPS), informasi posisi lubang dapat dikirimkan ke subsistem *website*. Pada *website*, data-data lokasi yang dikirimkan oleh mikrokontroller akan disusun sedemikian rupa untuk menampilkan antar muka kepada pihak pengelola agar mudah dipahami, dan dari *website* juga, apabila terdapat kendaraan yang akan melintasi lubang yang sebelumnya sudah terdata pada *database*, maka secara otomatis *website* akan mengirimkan data indikator yang berukuran kecil guna menyalakan modul *buzzer* pada mikrokontroller sebagai peringatan bahwa pengendara tersebut akan segera melintasi lubang.

3.2 Perangkat IoT

Sebelum berkuat pada perhitungan-perhitungan pada *website*, tentunya diperlukan rancangan perangkat IoT-nya untuk pengoperasiannya. Digunakan mikrokontroller NodeMCU ESP8266 yang dimana alat ini sudah tertanam modul WiFi sehingga dapat memudahkan akses internet dan tentunya lebih murah daripada menggunakan modul GSM. Gambar 5 merupakan ilustrasi dari alat yang akan digunakan.

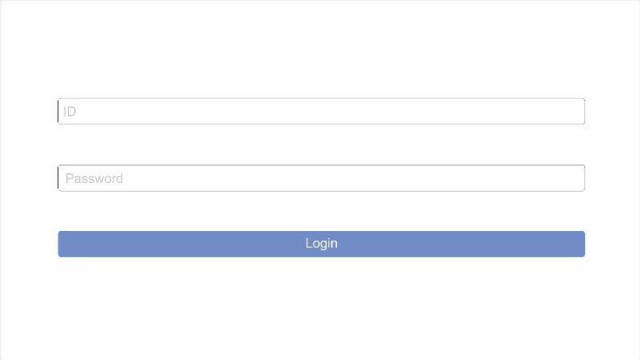


Gambar 5. NodeMCU ESP8266.

Mekanisme pengiriman data dengan NodeMCU ESP8266 adalah dengan mengirimkan nilai-nilai sensor yang sudah diolah ke *webserver* melalui suatu *link*. Metode ini disebut GET PHP, yaitu pembuatan *variable* baru melalui URL dengan nilai-nilai tertentu, dimana pada bagian *website* sendiri sudah menyiapkan pemeriksaan nama *variable* terbaru tersebut apakah sudah dibuat atau belum. Selanjutnya pengolahan akan dilanjutkan pada *webserver*, setelah diolah, *webserver* akan mengirimkan kembali hasil *response*-nya.

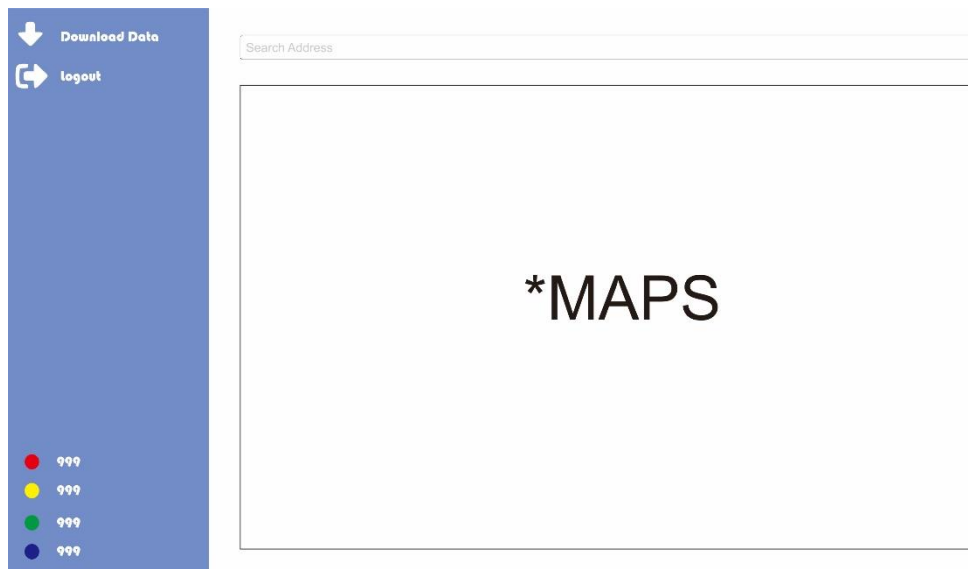
3.3 Desain Antar Muka Website

Halaman *website* terdiri dari ± 3 halaman, yaitu halaman *login*, halaman penampilan lokasi lubang dengan bantuan Bing Maps/Google Maps, serta halaman persetujuan untuk mendownload seluruh data lokasi yang ada. Berikut adalah perencanaan desain masing-masing halaman *website*.

A login form with a white background centered on a light gray background. It contains two text input fields: the top one is labeled 'ID' and the bottom one is labeled 'Password'. Below these fields is a solid blue button with the text 'Login' in white.

Gambar 6. Desain Halaman *Login*.

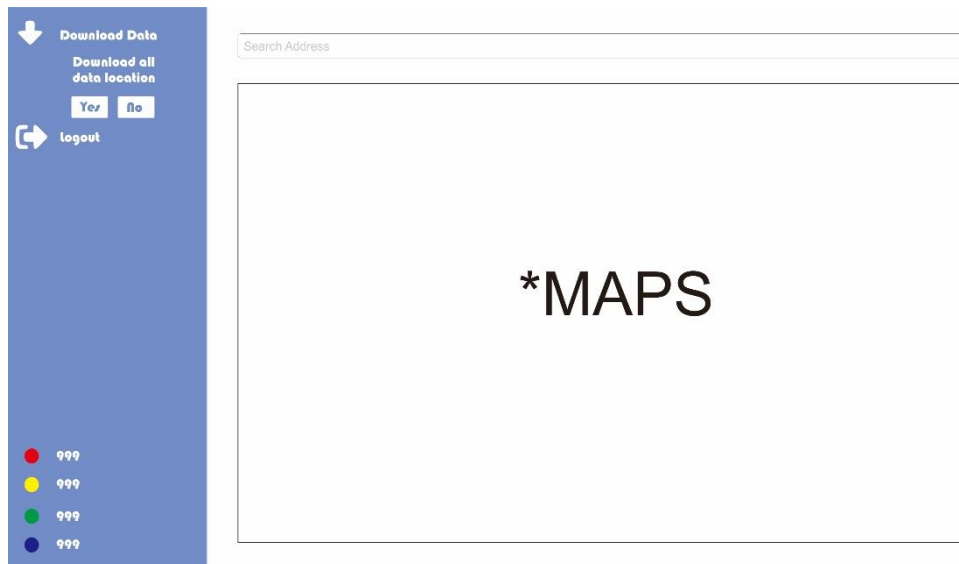
Gambar 6 merupakan desain halaman login. Halaman login hanya terdiri dari ID dan kata sandi, karena hanya pihak berwenang yang dapat mengakses dan mengelola *website* serta *database*-nya.

A web page design for a map display. On the left is a blue sidebar with a 'Download Data' button (downward arrow icon) and a 'logout' button (rightward arrow icon). Below these are four colored circles (red, yellow, green, blue) each followed by 'qqq'. The main content area has a search bar at the top labeled 'Search Address'. Below the search bar is a large rectangular placeholder for a map, with the text '*MAPS' centered inside it.

Gambar 7. Desain Halaman Penampilan Lubang.

Gambar 7 merupakan desain halaman penampilan lubang. Pada halaman ini akan dibutuhkan bantuan Bing Maps/Google Maps untuk menampilkan lokasi

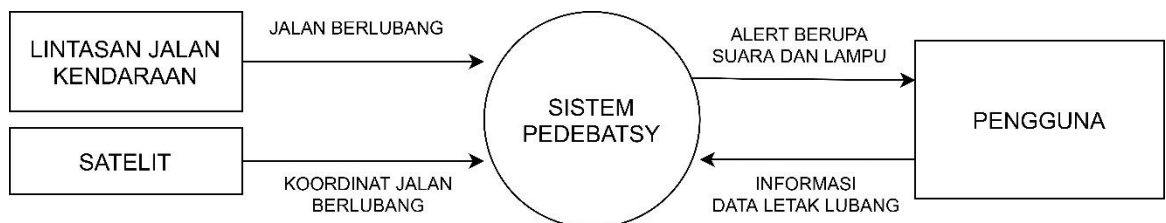
lubang melalui tampilan peta. Pada bagian atas peta terdapat fungsi untuk mencari suatu alamat pada Bing Maps/Google Maps untuk memeriksa apakah terdapat lubang disekitar alamat yang dicari. Fungsi ini bukan untuk tujuan mencari lokasi lubang.



Gambar 8. Desain Halaman *Download Data* Lokasi.

Seperti yang tampil pada Gambar 8, ketika diklik tombol “Download Data”, *website* akan meminta persetujuan untuk mengunduh semua data lokasi yang ada pada *database*. Berkas yang diunduh berupa *script* berkas lokasi koordinat lubang serta nilai *count* dari setiap *row* lubang.

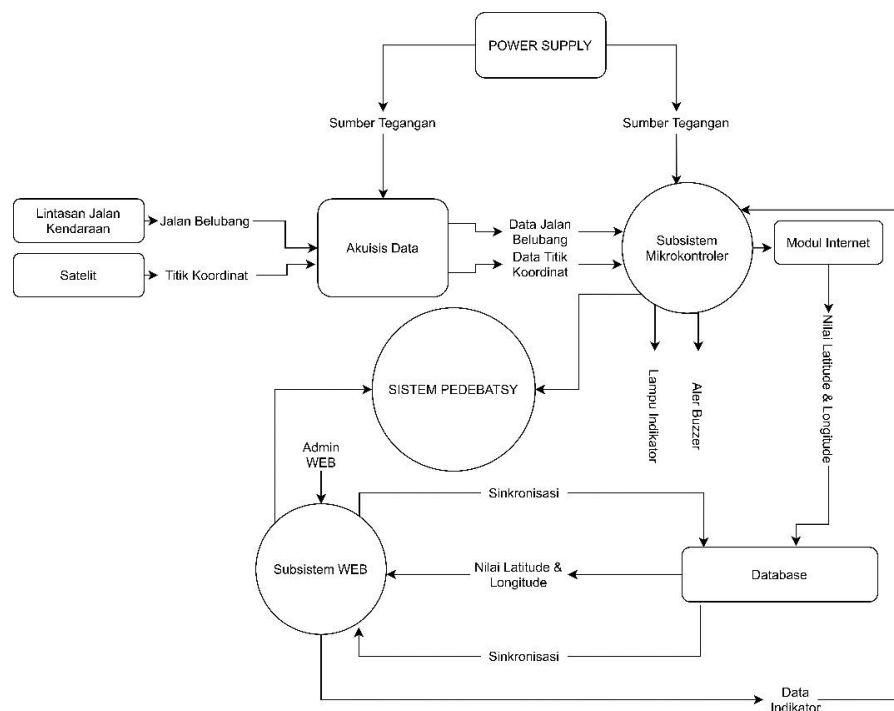
3.4 Mekanisme Cara Kerja Dari *Website*



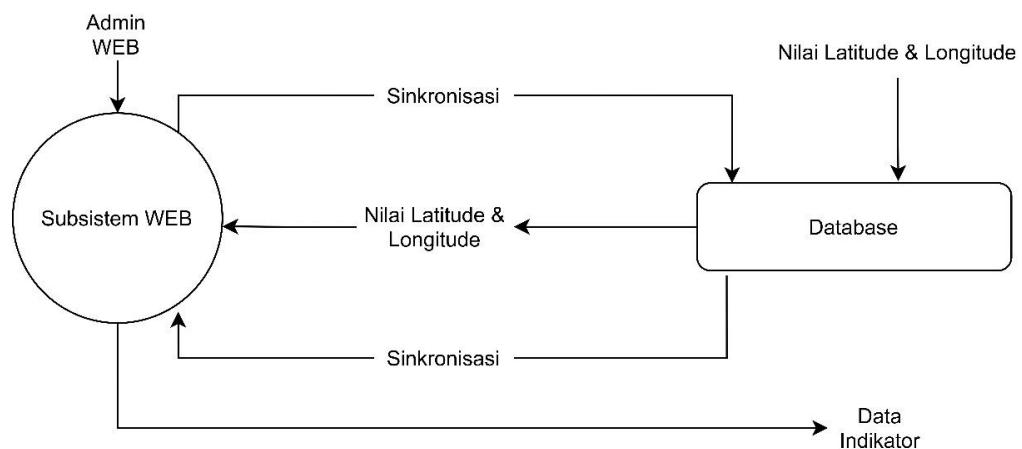
Gambar 9. DFD *Level 0*.

Berdasarkan *diagram flow data* (DFD) yang ditunjukkan pada Gambar 9, desain subsistem *website* merupakan *data center* dari data lokasi lubang yang dikirimkan oleh subsistem mikrokontroller. Dalam implementasinya dibutuhkan

bahasa pemrograman PHP dan MySQL untuk mengatur dan menyesuaikan posisi peta internet dengan data yang dikirimkan. Pada susbsistem ini, hanya dapat diakses oleh admin saja, yaitu pihak berwenang untuk memantau data yang ada sehingga pada halaman awal *website* akan ditampilkan antar muka berupa halaman *login* untuk memverifikasi apakah pengguna merupakan pihak berwenang atau tidak. Selain itu *webite* juga akan mengirimkan data indikator untuk memicu *buzzer* pada mikrokontroller berbunyi sebagai peringatan bahwa pengendara akan melewati lubang.



Gambar 10. DFD Level 1.



Gambar 11. DFD Level 2 Subsisitem Website.

Untuk memperjelas hubungan input dan output serta fungsi dari Gambar 10 dan Gambar 11, perhatikan tabel berikut:

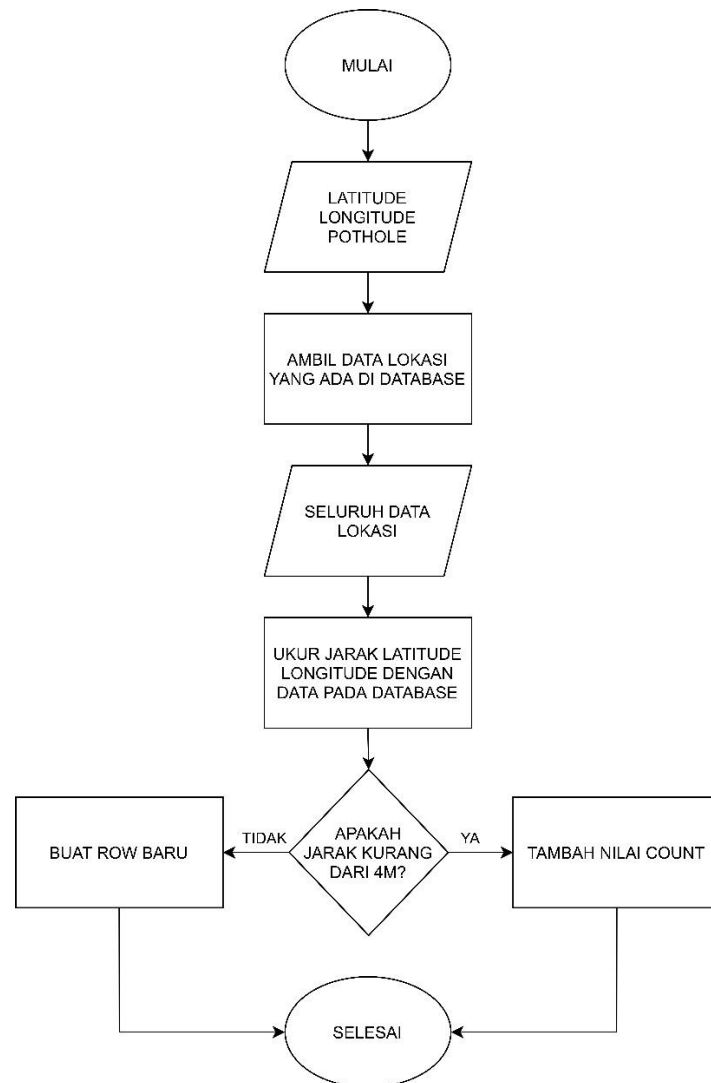
Tabel 1. Penjelasan DFD *Level 2* Subsistem *Website*.

Parameter	Keterangan
Input	• Data lokasi lubang apabila tervalidasi oleh mikrokontroller • Data GPS untuk menentukan arah dan kecepatan pengendara • <i>Username</i> dan kata sandi
Output	• Data indikator berukuran kecil yang dikirimkan ke mikrokontroller untuk menyalakan <i>buzzer</i> • Antarmuka <i>website</i> untuk pihak pengelola data lokasi
Fungsi	• Sinkronisasi data yang diterima dengan data yang sudah ada pada <i>database</i> • Pengiriman data indikator untuk memicu <i>buzzer</i> pada mikrokontroller menyala • Menampilkan antarmuka <i>website</i> kepada pihak pengelola agar data-data yang ada pada <i>database</i> dapat lebih mudah dimengerti

Pengguna alat ini akan secara otomatis mengirimkan titik lokasi terkini sebagai lokasi lubang apabila mikrokontroller mendeteksi adanya lubang melalui mekanisme pendeteksian lubang. Data yang dikirimkan oleh mikrokontroller akan diterima oleh website yang kemudian diolah untuk mensinkronisasikan data yang diterima dengan data yang sebelumnya sudah ada pada *website*. *Website* akan secara otomatis menghitung berapa kendaraan yang sudah melalui jalan tersebut untuk memvalidasi apakah lokasi yang dikirimkan layak untuk dianggap lubang yang perlu diwaspadai sehingga *website* perlu mengirimkan data indikator sebagai peringatan kepada pengendara lain kalau pengendara tersebut akan segera melewati lubang. Apabila *website* menerima data lokasi yang sama sebanyak 10x atau lebih, maka lokasi tersebut akan dijadikan sebagai lokasi lubang yang perlu diwaspadai sehingga *website* akan mengirimkan data indikator sebagai peringatan kepada pengendara lain yang akan melintasi lokasi tersebut.

Ada kalanya suatu jalan akan ditambal guna menghilangkan lubang di jalan, oleh karena itu, *website* akan melakukan sinkronisasi lagi secara otomatis untuk memvalidasi apakah data lokasi lubang yang ada sebelumnya masih layak dianggap sebagai lubang yang perlu diwaspadai atau tidak. Sehingga nantinya akan ada nilai *count* pada setiap lokasi jalan untuk menentukan apakah lokasi lubang tersebut masih layak sebagai lubang yang perlu diwaspadai atau tidak dengan menggunakan fungsi logika jika nilai *count* lebih besar atau sama dengan dari 10 maka dianggap sebagai lubang yang perlu diwaspadai, jika tidak maka bukan lubang yang perlu diwaspadai. Pada saat ada pengendara yang mengirimkan data lokasi lubang yang baru, maka akan ada nilai *count* baru yang bernilai 1.

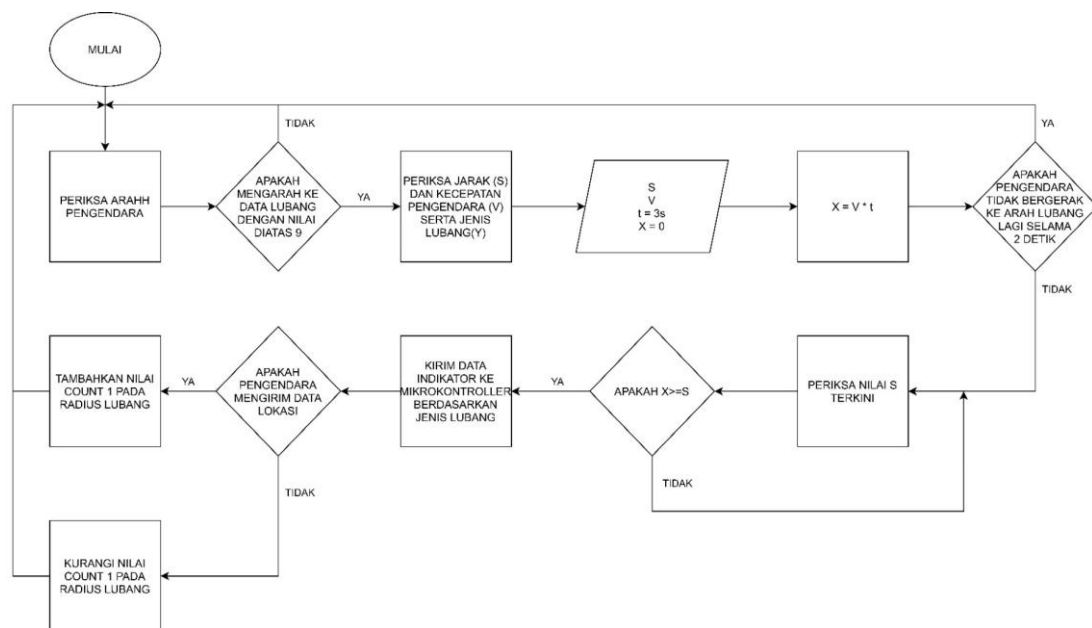
Setiap kali ada pengendara yang melintasi lubang dan mengirimkan data lokasi lubang yang sama, maka nilai *count* bertambah 1, dan setiap kali pengendara yang melewati lokasi lubang yang sebelumnya ada namun tidak mengirimkan data lokasi lubang, maka nilai *count* akan berkurang 1. Apabila nilai *count* sudah mencapai 0, *count* tersebut akan dihapus dari sistem. Dalam pengiriman data-data lokasi yang sama, tentunya data tersebut tidak selalu berada pada posisi yang sama persis satu sama lain, sehingga terdapat toleransi sebesar 4 meter dari jarak lokasi lubang satu-sama lain, supaya tidak terjadi pembuatan *count* baru yang terlalu banyak namun data lokasi lubang berdekatan. Gambar 12 merupakan *flowchart* dari mekanisme penerimaan data dari mikrokontroler.



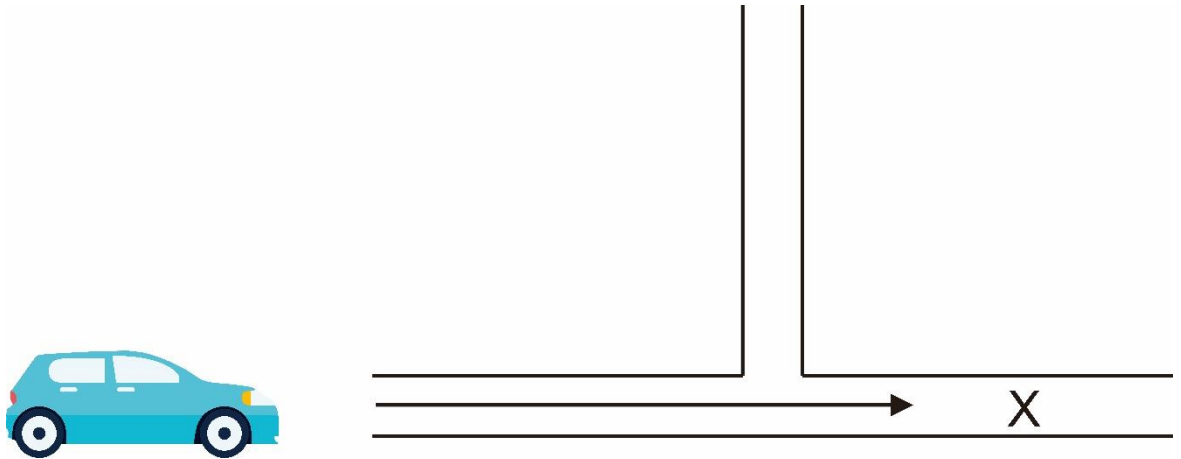
Gambar 12. *Flowchart* Mekanisme Penerimaan Data dan Sinkronisasi Data.

Dari data lokasi lubang yang sudah dikumpulkan sebelumnya, akan terdapat beberapa data lubang yang termasuk kategori lubang yang perlu diwaspadai ketika nilai *count* dari lubang tersebut bernilai lebih dari sama dengan 10. Mekanisme ini dijalankan dengan mengecek arah dari perubahan posisi melalui modul GPS dari mikrokontroller apakah pengendara sedang berjalan menuju lubang atau tidak. Ketika system mengkonfirmasi bahwa ada sebuah pengendara yang berjalan menuju lokasi lubang yang termasuk kategori lubang yang perlu diwaspadai, maka *website* akan mengirimkan data indikator ke mikrokontroller sebagai peringatan bahwa pengendara akan segera menemui lubang jalan yang perlu diwaspadai. Data indikator akan dikirimkan sekitar 3 detik sebelum pengendara melintasi lubang tersebut, sehingga jarak lubang yang dijadikan acuan untuk mengirimkan data

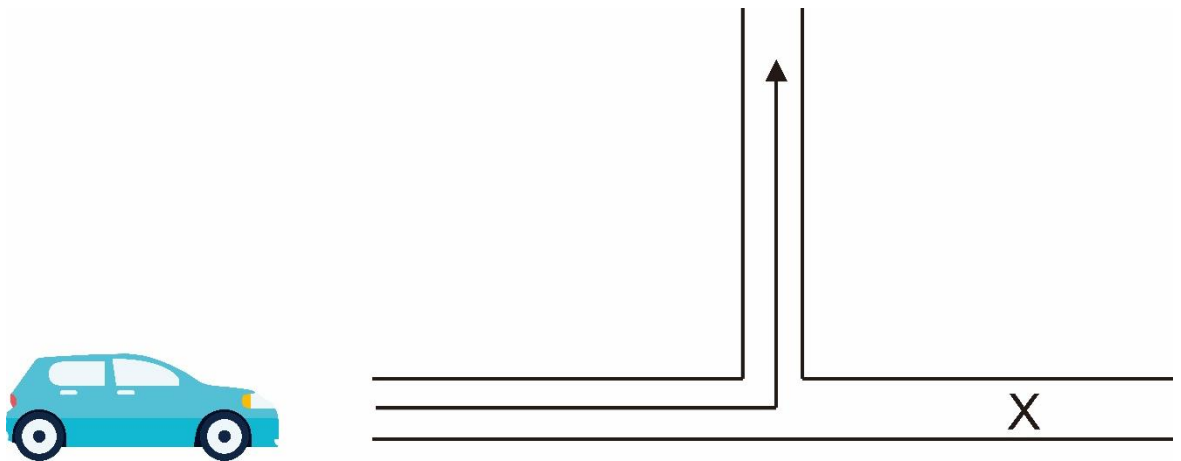
bergantung dari kecepatan pengendara tersebut. Akan tetapi, penentuan arah gerak pengendara ke lubang atau tidak akan dilakukan dalam jarak 100 meter atau kurang, karena apabila tidak dibatasi jarak tersebut, maka bisa saja ada lubang yang berjarak 2 km, namun system sudah bersiap-siap untuk mengirimkan data indikator, dan juga sistem akan menyiapkan fungsi apabila pengendara akan mengubah arah sehingga tidak jadi melintasi lubang serta tidak perlu mengirimkan data indikator. Pada mekanisme ini juga akan dilakukan sinkronisasi apakah pengendara mengirimkan data lokasi atau tidak. Jika iya, maka nilai *count* akan ditambah 1, sebaliknya jika pengendara tidak mengirimkan data lokasi. Data indikator ini berguna untuk menyalakan *buzzer* yang ada pada mikrokontroller sebagai peringatan tersebut. Gambar 13 merupakan *flowchart* dari mekanisme pengiriman data indikator sebagai peringatan:



Gambar 13. *Flowchart* Mekanisme Pengiriman Data Indikator.



Gambar 14. Pengendara Bergerak Menuju Lubang Dalam Jarak 100M.



Gambar 15. Pengendara Mengganti Arah Sehingga Tidak Lagi Mengarah Ke Lubang.

Gambar 14 dan Gambar 15 merupakan ilustrasi apabila pengendara merubah arahnya setelah system sudah bersiap untuk mengirimkan data indikator. Maka dalam kasus ini, berdasarkan *flowchart* yang ditampilkan pada Gambar 13 mekanisme pengiriman data indikator akan dibatalkan.

3.5 Struktur Tabel *Database*

Dari *database*, terdapat 2 data yang tersimpan, yaitu data lokasi dari lubang berserta jumlah *count*-nya yang tertera pada Tabel 3, serta data id dan kata sandi admin web untuk mengelola data lokasi yang tertera pada Tabel 2. Berikut merupakan penjelasan struktur tabel *database*:

1. Tabel Akun Admin Web

Nama Tabel : admin_web

Primary Key : id_admin

Fungsi : Menyimpan Data Akun

Tabel 2. Struktur Tabel Akun Admin Web.

No.	Field	Tipe Data	Constraint	Keterangan
1.	id_admin	int(3)	PK	Kode Admin
2.	id	var_char(20)	<i>Not Null</i>	Data untuk kolom ID
3.	kata_sandi	var_char(20)	<i>Not Null</i>	Data untuk kolom kata sandi

2. Tabel Data Lokasi Lubang

Nama Tabel : data_lokasi

Primary Key : id_lokasi

Fungsi : Menyimpan Data Lokasi Lubang

Tabel 3. Struktur Tabel Data Lokasi Lubang.

No.	Field	Tipe Data	Constraint	Keterangan
1.	id_lokasi	int(10)	PK	Kode Lokasi
2.	<i>latitude</i>	var_char(50)	<i>Not Null</i>	Informasi <i>latitude</i> lubang
3.	<i>longitude</i>	var_char(50)	<i>Not Null</i>	Informasi <i>longitude</i> lubang
4.	<i>pothole</i>	tinyint(1)	<i>Not Null</i>	Informasi mengenai tingkat kewaspadaan lubang
5.	<i>count</i>	var_char(20)	<i>Not Null</i>	Jumlah berapa kali lubang sudah terdeteksi
6.	<i>maxcount</i>	var_char(20)	<i>Not Null</i>	Jumlah count maximum yang pernah tercapai

7.	<i>waktu</i>	<code>var_char(30)</code>	<i>Not Null</i>	Informasi mengenai waktu <i>update</i> terakhir dari lubang tersebut
8.	<i>status</i>	<code>var_char(30)</code>	<i>Not Null</i>	Informasi apakah lubang sudah diperbaiki atau belum

3.6 Prosedur Pengujian

Untuk memvalidasi apakah sistem berkerja dengan baik, dibutuhkan proses pengujian dengan ketentuan toleransi tertentu. Berikut adalah beberapa pengujian yang akan dilakukan:

1. Pengujian Penerimaan Data Lokasi

Pengujian ini dilakukan dengan memasukan data sample dan melihat hasilnya pada database. Hasil dari pengujian ini adalah perbandingan letak posisi lubang secara *real* dengan letak posisi lubang secara data.

2. Pengujian Pengiriman Data Indikator

Pengujian ini dilakukan dengan melakukan data indikator berukuran kecil ke mikrokontroller untuk membunyikan *buzzer* sebagai peringatan apabila akan melintasi lubang. Hasil dari pengujian ini adalah kuisisioner mengenai ketepatan pemberian peringatan kepada pengendara.

3. Pengujian Antar Muka

Pengujian ini dilakukan untuk menguji *user experience* terhadap tampilan *website*. Hasil dari pengujian ini adalah kuisisioner mengenai pendapat apakah tampilan *website* sudah *user friendly* atau tidak.