

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Pada era sekarang ini, mobilitas yang tinggi dan cepat sudah merupakan suatu kebutuhan, tidak hanya bagi kaum pekerja namun juga bagi semua kalangan. Untuk mendukung mobilitas yang tinggi dibutuhkan alat transportasi yang dapat membantu pengguna untuk berpindah dari satu tempat ke tempat lain dengan waktu yang relatif lebih singkat. Dikarenakan kebutuhan akan alat transportasi yang sangat tinggi, pertumbuhan jumlah kendaraan baik itu kendaraan beroda dua atau beroda empat sangatlah pesat. Hal ini mengakibatkan kemacetan, selain dikarenakan banyaknya jumlah kendaraan yang menggunakan jalan, memarkirkan kendaraan di bahu jalan juga berdampak terjadi kemacetan. Untuk membantu mengurangi masalah kemacetan tersebut, maka lahan parkir sangat diperlukan untuk menampung kendaraan agar tidak mengganggu arus lalu lintas.

Universitas Katolik De La Salle Manado (UKDLSM) yang terletak di perbukitan Kombos [1] memiliki banyak mahasiswa dan karyawan yang menggunakan kendaraan milik pribadi, baik itu kendaraan roda empat (mobil) maupun kendaraan roda dua (motor). Selain itu, seringkali kampus mengadakan kegiatan-kegiatan yang dihadiri oleh tamu dari luar kampus yang juga meningkatkan jumlah kendaraan sehingga kapasitas lahan parkir akan lebih cepat penuh. Jumlah kepadatan kendaraan yang cukup banyak membuat lahan parkir menjadi penuh sehingga para pengendara sulit untuk mencari tempat parkir. Informasi yang minim terhadap keadaan lahan parkir di UKDLSM, mengakibatkan kesulitan bagi para pengguna lahan parkir untuk memprediksi apakah lahan parkir sudah penuh atau belum.

*Internet of Things* (IoT) merupakan suatu konsep di mana objek tertentu memiliki kemampuan untuk mengirim data yang diperoleh melalui sensor dan jaringan internet secara otomatis [2]. Untuk mendukung teori dari penerapan IoT ini maka diambil beberapa contoh penelitian yang berhubungan dengan judul yang diangkat. Pada penelitian pertama [3], membuat aplikasi yang mampu menghitung jumlah kendaraan di tempat parkir dengan menganalisis jumlah kendaraan yang

masuk/keluar pada area parkir terbuka secara *real time* dengan menggunakan pengolahan citra. Pada penelitian kedua [5], membuat pembayaran pada gerbang tol menjadi lebih efisien, dengan membuat aplikasi berbasis IoT, di mana setiap mobil dipasang *receiver*. Proses transaksi ini dapat dilakukan pada radius 10 meter dari gerbang tol tanpa melakukan pemberhentian dengan kecepatan rata-rata 10 km/jam. Pada penelitian ketiga [6], dibuatkan aplikasi untuk mendeteksi tempat parkir yang kosong berbasis IoT dengan menggunakan sensor *ultrasonic* sebagai sensor pendeteksi objek yang kemudian diolah menjadi informasi untuk menemukan tempat parkir yang kosong.

Berdasarkan uraian masalah yang ada, dibuatlah aplikasi penghitung kendaraan berbasis IoT untuk menghitung banyaknya kendaraan yang masuk dan keluar. Kendaraan harus melewati portal yang telah dilengkapi oleh sensor, yang akan mendeteksi setiap kendaraan yang lewat. Untuk membantu mengklasifikasikan data yang diperoleh, maka penulis akan menggunakan algoritma C4.5 untuk membantu menganalisis pola-pola data dan mengklasifikasikannya ke dalam 3 *class*, yaitu mobil, motor dan orang. Melalui hasil klasifikasi yang diperoleh maka dapat memberikan informasi tentang jumlah kendaraan yang berada di dalam kampus. Ketika lahan parkir kampus telah penuh, maka staf yang bertugas akan mengarahkan pengendara untuk memarkirkan kendaraan di luar kampus. Dengan begitu, aplikasi ini dapat memberikan informasi keadaan tempat parkir sehingga mahasiswa dan karyawan dapat mengantisipasi jika tempat parkir di UKDLSM sudah penuh.

## 1.2 Rumusan Masalah

Bagaimana membangun sebuah Aplikasi Penghitung Kendaraan di Universitas Katolik De La Salle Manado Berbasis *Internet of Things* yang dapat menghitung kendaraan dan memberikan informasi bagi pengendara jika lahan parkir telah penuh?

### 1.3 Tujuan Tugas Akhir

Membangun Aplikasi Penghitung Kendaraan di Universitas Katolik De La Salle Manado Berbasis *Internet of Things* yang dapat memberikan informasi lahan parkir bagi pengendara ketika lahan parkir telah penuh.

### 1.4 Manfaat Tugas Akhir

Berikut ini merupakan manfaat dari tugas akhir:

1. Meningkatkan kemampuan dalam membuat dan merancang aplikasi sesuai dengan kebutuhan pengguna.
2. Memberikan informasi tentang status tempat parkir pada UKDLSM.

### 1.5 Batasan Masalah

Batasan masalah dalam pembuatan aplikasi yang akan dibangun, yaitu:

1. Aplikasi hanya akan diuji di Universitas Katolik De La Salle Manado.
2. Alat yang dipasang hanya dua, yaitu dipasang pada portal masuk kampus dan portal keluar kampus.
3. Klasifikasi dilakukan terhadap semua objek yang mungkin lewat di portal jalan masuk dan keluar tempat parkir, seperti mobil, motor dan orang. Meskipun aplikasi hanya akan menghitung jumlah kendaraan yang masuk dan keluar, klasifikasi orang dilakukan sebagai pengenalan objek agar tidak terjadi kesalahan klasifikasi.
4. Pada tugas akhir ini, untuk dapat mendeteksi dua objek, penulis menggunakan asumsi bahwa objek kedua bergerak melewati sensor minimal 1 detik setelah objek pertama melewati sensor.
5. Sensor yang digunakan yaitu sensor *ultrasonic* HC-SR04 dan sensor *accelerometer gyroscope* MPU9266.
6. Menggunakan algoritma C4.5.
7. Fitur yang dikembangkan yaitu
  - a. Menghitung jumlah kendaraan yang masuk dan keluar.
  - b. Memberikan informasi status tempat parkir di UKDLSM.
  - c. Membuat laporan jumlah keluar masuk kendaraan.

## 1.6 Metodologi Pengembangan Aplikasi

Metodologi pengembangan perangkat lunak yang akan digunakan dalam membangun aplikasi yaitu *Scrum*. *Scrum* merupakan metodologi yang menggunakan prinsip Agile yang populer, dikarenakan metode ini lebih adaptif, cepat, fleksibel dan efektif [4]. Metodologi *Scrum* memiliki 5 tahapan, yaitu [4]:

1. *Initiate*: Pada tahap ini akan dilakukan analisis *requirement* dari aplikasi yang akan dibangun. Tujuan utama dari tahap ini yaitu untuk melakukan perancangan sistem dan *project vision*.
2. *Plan and Estimate*: Tahap ini terdiri dari proses yang berkaitan dengan tugas perencanaan dan estimasi dari aplikasi yang akan dibangun,
3. *Implement*: Tahap ini merupakan tahapan untuk mengeksekusi setiap tahapan yang telah dirancang sebelumnya dan melakukan aktivitas-aktivitas untuk membuat produk.
4. *Review and Retrospect*: Tahap ini berkaitan dengan *review* hasil kerja dan pekerjaan yang telah dilakukan sebelumnya. Hal ini bertujuan untuk mencari tahu kekurangan dari aplikasi yang dibangun.
5. *Release*: Pada tahap ini produk yang telah melewati persyaratan, analisis, perancangan dan implementasi, telah siap diluncurkan. Selain itu juga dilakukan dokumentasi mengenai pelajaran apa saja yang diperoleh selama melakukan pengembangan.

## 1.7 Sistematika Penulisan

Berikut merupakan sistematika penulisan dari tugas akhir:

**BAB I PENDAHULUAN:** Menjelaskan tentang latar belakang, rumusan masalah, tujuan, manfaat tugas akhir, batasan masalah, metodologi penulisan dan sistematika penulisan dari pelaksanaan tugas akhir.

**BAB II STUDI PUSTAKA:** Menjelaskan tentang seluruh teori pendukung yang akan digunakan dan metodologi pengembangan aplikasi.

**BAB III ANALISIS:** Pada bagian ini akan dilakukan analisis terhadap kebutuhan, proses bisnis, dan metode pemecahan masalah yang akan digunakan dalam aplikasi.

**BAB IV PERANCANGAN:** Bagian ini menjelaskan tentang kaskas pemodelan dan desain arsitektur aplikasi.

**BAB V IMPLEMENTASI:** Bagian ini menjelaskan tentang penerapan hasil dari perancangan yang dilakukan pada tahap-tahap sebelumnya.

**BAB VI PENGUJIAN:** Bagian ini menjelaskan tentang hasil dari pengekseskuan aplikasi yang telah melalui tahap pengujian.

**BAB VII KESIMPULAN DAN SARAN:** Membahas tentang kesimpulan dari pembahasan aplikasi yang dibangun dan saran untuk pengembangan aplikasi di kemudian hari.

## BAB II STUDI PUSTAKA

Pada bab ini akan membahas tentang teori-teori yang akan digunakan dalam pembuatan aplikasi penghitung kendaraan berbasis IoT. Teori-teori pendukung diperoleh dari buku, jurnal, artikel dan internet sebagai sumber yang digunakan.

### 2.1 *Internet of Things*

*Internet of Things* atau biasa disingkat IoT, pertama kali dicetuskan oleh Kevin Ashton pada salah satu presentasinya di tahun 1999. Sejak saat itu banyak perusahaan besar mulai mendalami IoT, seperti *Microsoft*, *Intel*, *Oracle* dan masih banyak lagi. Hal ini dikarenakan IoT menawarkan banyak potensi yang dapat mengubah dunia. Melihat dari arti kata *Internet of Things* itu sendiri adalah internet dari peralatan-peralatan/perangkat-perangkat. IoT adalah bagaimana internet dapat berkomunikasi dengan perangkat-perangkat yang ada dan dapat bermanfaat [7].

IoT merupakan suatu konsep di mana objek tertentu memiliki kemampuan untuk mentransfer data yang diperoleh melalui sensor dan jaringan internet secara otomatis [2]. IoT bertujuan untuk memperluas manfaat dari konektivitas internet yang terhubung secara terus menerus. Kemajuan infrastruktur internet sangat berdampak bagi dunia teknologi, di mana bukan hanya komputer atau *smartphone* saja yang dapat terkoneksi ke internet, namun benda-benda seperti sensor, mesin produksi, peralatan elektronik, dan benda-benda lain dapat terhubung juga [8].

### 2.2 **Algoritma C4.5**

Algoritma C4.5 merupakan metode yang digunakan untuk masalah klasifikasi dalam *machine learning* dan *data mining*, yang bertujuan untuk pengelompokan dan bersifat prediksi untuk menemukan pola dengan nilai yang relatif besar hingga sangat besar. Algoritma C4.5 merupakan pengembangan dari algoritma ID3 atau mempunyai prinsip dasar kerja sama dengan algoritma ID3. Ada beberapa perbedaan antara algoritma ID3 dan algoritma C4.5 di mana kekurangan

pada algoritma ID3 dapat ditutupi oleh algoritma C4.5 yaitu dapat menangani variabel yang bertipe diskrit maupun kontinu, dapat menangani variabel yang memiliki *missing*, dapat memangkas cabang pohon keputusan.

Algoritma C4.5 dipilih untuk membantu pengklasifikasian objek pada aplikasi penghitung kendaraan berbasis IoT, karena aplikasi ini akan mengklasifikasikan objek sehingga diperlukan sebuah algoritma yang dapat membantu proses tersebut. Oleh karena itu algoritma yang cocok adalah *decision tree*, karena algoritma ini dapat mengklasifikasikan objek dalam bentuk pohon keputusan dan cocok dengan bentuk data diskrit yang mana data yang akan diklasifikasikan pada aplikasi juga berbentuk diskrit. Ada beberapa algoritma alternatif yang dapat digunakan dalam pembentukan pohon keputusan, tetapi dalam tugas akhir ini dipilih algoritma C4.5 yang merupakan pengembangan dari algoritma ID3.

Secara umum algoritma C4.5 memiliki langkah-langkah dalam membangun pohon keputusan [22], yaitu:

1. Mencari atribut yang akan menjadi *root*.
2. Buat cabang untuk masing-masing nilai.
3. Bagi kasus dalam cabang
4. Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Untuk memilih atribut atau variabel sebagai akar, dapat diperoleh dari hasil *gain* tertinggi yang didapat dengan menggunakan rumus di bawah ini.

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i) \dots\dots\dots(2.1)$$

Di mana, S merupakan himpunan kasus, sedangkan atribut yang disimbolkan dengan A dan n merupakan jumlah partisi dari atribut A. Selanjutnya jumlah kasus pada partisi ke-I yaitu  $|S_i|$  dan jumlah kasus dalam S adalah  $|S|$ . Sebelum menghitung *Gain*, maka harus menghitung entropy untuk menentukan seberapa informatif sebuah masukan atribut. Berikut merupakan rumus untuk menghitung entropy yang dapat dilihat pada halaman selanjutnya.

$$Entropy(S) = \sum_{i=1}^n -p_i * \log_2 p_i \dots\dots\dots(2.2)$$

Keterangan:

S : himpunan kasus.

n : jumlah kelas klasifikasi.

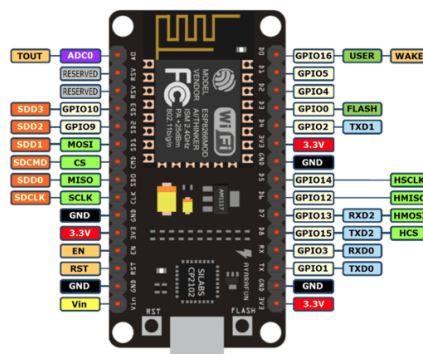
$p_i$  : proporsi dari  $S_i$  terhadap S.

## 2.3 Teknologi yang Digunakan

Adapun teknologi yang digunakan untuk membangun aplikasi penghitung kendaraan berbasis IoT terbagi menjadi dua, yaitu *hardware* dan *software*. *Hardware* yang digunakan adalah NodeMCU dan sensor-sensor seperti *Ultrasonic* dan *Accelerometer*.

### 2.3.1 NodeMCU ESP8266

NodeMCU ESP8266 merupakan modul pengembangan dari modul *platform* IoT yang bersifat *open source* dan pengembangan *kit* menggunakan bahasa pemrograman LUA atau dapat menggunakan Arduino IDE. ESP8266 yang digunakan saat ini adalah versi 1.0 yang merupakan pengembangan dari versi 0.9. Versi 1.0 ini menggunakan tipe ESP-12E yang merupakan pengembangan dari ESP-12 pada versi sebelumnya. Modul WiFi NodeMCU adalah *firmware* interaktif berbasis LUA Espressif ESP8622 Wifi SoC. Kelebihan dari NodeMCU ini yaitu *board* yang berukuran kecil dengan panjang 4,83 cm, lebar 2,54 cm dan berat 7 gram sehingga penggunaan NodeMCU ini lebih efisien dari segi biaya dan tempat [18]. Untuk melihat bentuk fisik dari NodeMCU ESP8266 dapat dilihat pada Gambar 2.1.



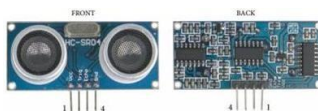
Gambar 2.1 NodeMCU ESP8266 [18]



### 2.3.2 Sensor Ultrasonik

Sensor ultrasonik atau HC-SR04 merupakan sebuah sensor yang berfungsi untuk mengubah bunyi menjadi listrik dan sebaliknya. Frekuensi yang dihasilkan oleh sensor ini adalah 20.000 Hz. Bunyi dari sensor ultrasonik ini tidak dapat ditangkap oleh pendengaran manusia. Cara kerja sensor ini yaitu, alat ini akan menembakkan gelombang ultrasonik ke suatu area yang telah ditargetkan. Setelah gelombang sampai pada target, maka gelombang akan memantul kembali ke arah sensor. Gelombang pantulan tersebut akan ditangkap oleh sensor, kemudian akan dihitung selisih waktu dari saat sensor mengirimkan gelombang sampai sensor menerima gelombang kembali, maka akan diperoleh hasil berupa jarak [10].

HC-SR04 merupakan sensor ultrasonik yang dapat mengukur jarak benda mulai dari 2 cm hingga 4 meter dengan akurasi 3mm. Sensor ini memancarkan gelombang ultrasonik melalui udara. Kekurangan sensor ini adalah jika ada objek yang menghalangi gelombang maka akan langsung memantul kembali sehingga hasilnya menjadi tidak akurat [10]. HC-SR04 dapat dilihat pada Gambar 2.2.



**Gambar 2.2 Sensor Ultrasonik HC-SR04 [10]**

### 2.3.3 Sensor Accelerometer

*Accelerometer* atau MPU 9266 adalah sebuah sensor yang berfungsi untuk mengukur percepatan suatu objek. Sensor ini dapat mengukur percepatan dinamis dan statis, serta dapat mengukur percepatan yang disebabkan karena gravitasi. Sensor *accelerometer* mengukur percepatan yang disebabkan oleh gerakan objek yang melekat padanya. Pendeteksian percepatan gerakan ini ada 3 sumbu yaitu kanan-kiri, atas-bawah dan depan-belakang. *Accelerometer* ini dapat digunakan seperti mengukur getaran pada kendaraan, benda padat, mesin, dan masih banyak lagi. Cara kerja dari sensor ini adalah dengan prinsip percepatan [11].



**Gambar 2.3 Sensor Accelerometer [11]**

## 2.4 Bahasa Pemrograman

Adapun bahasa pemrograman yang akan digunakan dalam membangun aplikasi penghitung kendaraan berbasis *Internet of Things* yaitu PHP dan *JavaScript*.

### 2.4.1 PHP

*Hypertext Preprocessor* (PHP) adalah bahasa pemrograman dan *interpreter* yaitu penerjemah baris kode sumber menjadi kode mesin sehingga dapat dimengerti oleh komputer secara langsung pada saat kode dijalankan. PHP merupakan *script* yang bertujuan untuk membuat *website* yang dinamis [12]. Semua *script* PHP dieksekusi pada *server*, sebenarnya untuk membuat sebuah *website*, tidak harus menggunakan PHP. *Website* dapat dibuat hanya dengan HTML dan CSS saja. *Website* yang dibangun menggunakan HTML dan CSS ini dikenal sebagai *website* statis, di mana isi konten dari *website* ini tidak akan berubah. Sedangkan, *website* dinamis ini memungkinkan konten bisa menyesuaikan tergantung situasi dan dapat berubah-ubah sesuai dengan *script* yang dijalankan. *Website* dinamis bisa dibuat menggunakan PHP. Contohnya dengan menggunakan salah satu fungsi logika pada PHP yaitu *if/else*. Dengan fungsi ini halaman *web* akan lebih dinamis, karena hasil yang ditampilkan akan tergantung kondisi. *Website* dinamis juga dapat menerima dan menyimpan data ke dalam basis data, membuat halaman yang berubah-ubah sesuai input dari *user*, memproses *form* dan sebagainya [13]. Versi PHP yang digunakan dalam pembangunan aplikasi ini adalah versi 7.

### 2.4.2 JavaScript

*JavaScript* merupakan bahasa *script* populer yang digunakan untuk membuat halaman web yang dapat merespons *event* yang terjadi pada halaman secara langsung tanpa harus memuat ulang halaman seperti kasus pada PHP. *JavaScript* berfungsi untuk menyatukan halaman-halaman web. *JavaScript* merupakan bahasa sisi klien yang didesain pada *browser* komputer pengguna, bukan pada *server*. Program *JavaScript* digunakan untuk beraksi terhadap *event* yang dibuat oleh pengguna. *JavaScript* dapat digunakan untuk mengendalikan tampilan halaman yang biasa dikenal dengan sebutan DOM (*Document Object*

*Model*). *JavaScript* berperan sebagai bahasa pemrograman, yang memiliki struktur dasar seperti tipe data, *variable*, *look control*, *statement if/else*, *statement switch*, fungsi, dan objek [14].

## 2.5 MySQL

*My Structured Query Language* (MySQL) merupakan salah satu dari sekian banyak *Database Management System* (DBMS) lain seperti *Oracle*, *Microsoft SQL Server*, *ADABAS* dan lain sebagainya. MySQL berfungsi untuk mengolah basis data menggunakan bahasa *Structured Query Language* (SQL). MySQL bersifat *open source* yang berarti dapat kita gunakan secara gratis [13]. SQL merupakan suatu bahasa permintaan yang terstruktur. SQL adalah Bahasa yang dipergunakan untuk mengakses data dalam basis data. Bahasa ini merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini, hampir semua *server* basis data yang ada mendukung bahasa ini untuk menjalankan manajemen datanya. SQL juga diartikan sebagai antarmuka standar untuk sistem manajemen basis data relasional. SQL memiliki beberapa kumpulan perintah yang dapat digunakan untuk membuat, mengubah struktur serta definisi tipe data dari basis data seperti tabel, index, dan lain-lain [17]. Kumpulan perintah ini disebut *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). Berikut penjelasannya:

### 2.5.1 *Data Definition Language* (DDL)

*Data Definition Language* (DDL) adalah kumpulan perintah SQL yang digunakan untuk membuat sebuah basis data. Dapat dikatakan *Data Definition Language* (DDL) adalah perintah untuk menggambarkan desain dari basis data secara menyeluruh dan membuat struktur dasar dari database dan tabel. DDL juga dapat menciptakan, mengubah, dan menghapus basis data dan tabel dengan perintah-perintah DDL. Seorang pengolah basis data pasti sering menggunakan perintah DDL apa lagi dalam membuat dan mengubah struktur tabel sebab sering kali tabel yang dibuat harus dilakukan perubahan dikarenakan harus disesuaikan dengan masukan data yang ada dan juga aplikasi yang dibuat *programmer*.

Perintah-perintah yang termasuk DDL adalah statemen *connect*, *create*, *show*, *describe*, *drop*, *rename* dan *alter* [16].

### 2.5.2 Data Manipulation Language (DML)

DML adalah pengambilan informasi yang tersimpan pada basis data. Pemberian informasi baru pada basis data, penghapusan pada basis data, modifikasi basis data. Dapat dikatakan DML adalah sebuah bahasa yang memungkinkan pengguna untuk mengakses dan memanipulasi data yang terorganisasi oleh model data yang tepat. Pada dasarnya DML memiliki 2 jenis yaitu prosedural dan non prosedural. Prosedural DML membutuhkan pengguna untuk menspesifikasikan data apa yang dibutuhkan dan bagaimana cara mendapatkannya, sedangkan non prosedural DML membutuhkan pengguna untuk menspesifikasikan data apa yang dibutuhkan tanpa tahu bagaimana cara mendapatkannya [16].

## 2.6 Metodologi Pengembangan Sistem

Adapun metodologi pengembangan sistem yang akan digunakan dalam membangun aplikasi penghitung kendaraan berbasis IoT yaitu metodologi *Scrum*. Untuk pemodelan sistem akan digunakan *Flowchart* dan *Entity Relationship Diagram* (ERD).

### 2.6.1 Scrum

*Scrum* merupakan metodologi yang menggunakan prinsip Agile yang populer dikarenakan metode ini lebih adaptif, cepat, fleksibel dan efektif [4]. *Scrum* diperkenalkan pertama kali oleh Jeff Sutherland pada awal tahun 1990-an. *Scrum* merupakan *framework* yang bertujuan untuk membantu menyelesaikan masalah yang kompleks dan berubah-ubah dan secara bersamaan dapat menghasilkan produk dengan nilai setinggi mungkin. Pada metodologi *scrum* ada istilah kegiatan *scrum* atau biasa disebut *sprint*. *Sprint* merupakan batasan waktu untuk menyelesaikan potongan produk yang akan dibuat. Tahapan satu *sprint* terdiri dari *sprint planning*, *daily scrum*, *development*, *sprint review* dan *sprint retrospective* [20]. Berikut merupakan tahapan-tahapan dari metodologi *scrum* [4], yaitu:


1. *Initiate*: Pada tahap ini akan dilakukan analisis *requirement* dari aplikasi yang akan dibangun. Tujuan utama dari tahap ini yaitu untuk melakukan perancangan sistem dan *project vision*.
2. *Plan and Estimate*: Tahap ini terdiri dari proses yang berkaitan dengan tugas perencanaan dan estimasi dari aplikasi yang akan dibangun.
3. *Implement*: Tahap ini merupakan tahapan untuk mengimplementasi setiap tahapan yang telah dirancang sebelumnya, dan melakukan aktivitas-aktivitas untuk membuat produk. Pada tahap ini akan dilakukan proses pembangunan aplikasi dengan menggunakan bahasa pemrograman PHP sebagai *front-end* dan *web service REST API* sebagai *back-end*.
4. *Review and Retrospect*: Tahap ini berkaitan dengan *review* hasil kerja dan pekerjaan yang telah dilakukan sebelumnya. Hal ini bertujuan untuk mencari tahu kekurangan dari aplikasi yang dibangun. Pada tahap ini penulis melakukan pengujian alat dan program di portal masuk/keluar UKDLSM yang bertujuan untuk meninjau dan mengetahui kekurangan dari aplikasi yang dibangun.
5. *Release*: Pada tahap ini produk yang telah melewati persyaratan, analisis, perancangan dan implementasi, telah siap diluncurkan. Selain itu juga akan dilakukan dokumentasi mengenai apa saja yang diperoleh selama ini.




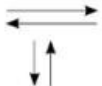
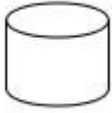
Pada tugas akhir ini tidak membahas tahap kelima yaitu *release*, karena aplikasi hanya digunakan untuk keperluan tugas akhir.

### 2.6.2 Flowchart

Diagram ini menunjukkan susunan prosedur. Diagram ini menyatakan perbedaan antara aktivitas untuk digabungkan menjadi sebuah kesatuan sistem [17]. Pada diagram *flowchart* terdapat beberapa simbol yang akan digunakan, dilihat pada Tabel 2.1.

**Tabel 2.1 Flowchart [17]**


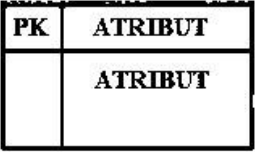


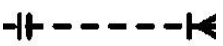
<b>Nama</b>	<b>Simbol</b>	<b>Keterangan</b>
<i>Terminal</i>		Simbol untuk menunjukkan awal ( <i>start</i> ) dan akhir ( <i>stop</i> ).

Nama	Simbol	Keterangan
<i>Process</i>		Simbol <i>flowchart</i> yang berfungsi untuk menunjukkan pengolahan, langkah atau tindakan.
<i>Input/Output</i>		Bentuk ini menunjukkan masukan ( <i>input</i> ) atau keluaran ( <i>output</i> ) dari suatu proses.
<i>Decision</i>		Simbol <i>decision</i> ini berfungsi sebagai simbol percabangan keputusan.
<i>Flow</i>		simbol ini berfungsi sebagai alur yang menghubungkan antara simbol satu dengan simbol yang lain atau menyatakan jalannya arus dari suatu proses.
<i>Database</i>		Silinder merupakan simbol yang digunakan untuk menggambarkan basis data yang melambangkan data yang disimpan dalam <i>hard drive</i> .

### 2.6.3 Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* (ERD) merupakan teknik pemodelan data yang menunjukkan hubungan logis dan interaksi antara entitas sistem. ERD memberikan tampilan keseluruhan sistem dan *blueprint* untuk membuat struktur data fisik [21]. Pada tugas akhir ini menggunakan ERD versi Chen yang memiliki beberapa simbol yang dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol ERD [21]

Nama	Nama Simbol	Keterangan
<i>Entity</i>		Simbol ini merupakan kumpulan-kumpulan objek yang dapat diidentifikasi secara unik atau berbeda. Simbol entitas ini biasanya digambarkan dengan persegi panjang.
<i>Attribute</i>		Simbol atribut berfungsi untuk mendeskripsikan entitas tersebut. Biasanya atribut memiliki <i>primary key</i> (kunci utama).
<i>Relation</i>		<i>One to one</i> , setiap entitas hanya bisa berelasi dengan satu entitas lain.
		<i>Many to many</i> , setiap entitas bisa mempunyai relasi dengan entitas lain, dan sebaliknya.
		<i>One to many</i> , hubungan antara satu entitas dengan beberapa entitas dan sebaliknya.

## 2.7 Ulasan Penelitian Terkait

Pada bagian ini akan membahas tentang ulasan penelitian yang serupa dengan judul tugas akhir yang diangkat sebagai berikut:

### a. Perancangan Sistem Penghitung Jumlah Kendaraan Pada Area Parkir Dengan Metode *Background Subtraction* Berbasis *Internet Of Things* [3]

Penelitian ini membuat aplikasi yang mampu menghitung jumlah kendaraan di tempat parkir dengan cara menganalisis jumlah kendaraan yang masuk/keluar pada area parkir terbuka secara *real time*, untuk mendeteksi, menghitung dan mengklasifikasikan objek. Penulis menggunakan pengolahan citra dengan metode *background subtraction* dengan OpenCV. Proses deteksi gerak

berfungsi untuk mendeteksi objek bergerak dari video yang direkam oleh kamera. Deteksi gerak menggunakan fungsi yang terdapat pada library OpenCV, melalui API (*Application Programming Interface*) C++ bernama `cv2.BackgroundSubtractorMOG`. Fungsi tersebut memiliki parameter di antaranya *history*, *varThreshold*, dan *bShadowDetection*. Parameter histori berfungsi untuk menetapkan jumlah *frame* terakhir yang mempengaruhi model latar belakang. Parameter *varThreshold* berfungsi untuk mengatur nilai ambang pada *background subtraction*. Parameter *bShadowDetection* berfungsi untuk mendeteksi bayangan pada video.

Parameter lain yang harus diatur, yakni *learningRate*. Parameter *learningRate* mengatur nilai laju pembelajaran yang akan digunakan. Pada perancangan ini, parameter *learningRate* akan diatur sesuai dengan kondisi lingkungan. Nilai parameter *learningRate* pada kondisi lingkungan sistem akan ditentukan dari hasil pengujian variasi parameter *learningRate*. Proses klasifikasi jenis kendaraan digunakan untuk membedakan jenis kendaraan yang terdeteksi, yaitu mobil atau motor. Klasifikasi jenis kendaraan menggunakan data koordinat nilai tinggi dan data koordinat nilai lebar untuk menentukan ukuran kendaraan yang terdeteksi oleh sistem [3].

**b. Sistem Deteksi Mobil pada Aplikasi Pembayaran Gerbang Tol Berbasis *Internet of Things* [5]**

Untuk mengurangi jumlah kemacetan yang semakin meningkat, pemerintah dan pihak pengelola jalan tol telah membangun beberapa ruas jalan tol sebagai alternatif dan menerapkan pembayaran transaksi secara elektronik berbasis RFID (*Radio Frequency Identification*). Sesuai dengan Standar Pelayanan Minimal (SPM) yang diatur pada Permen PU No. 16/PRT/M/2014, waktu pembayaran transaksi pada gerbang tol otomatis dilakukan maksimal selama 5 detik untuk setiap kendaraan. Namun, pembayaran dengan menggunakan RFID belum mampu secara efektif mengurangi antrian kemacetan di gerbang pembayaran tol. Untuk mengurangi jumlah kemacetan yang semakin meningkat, dibangun sistem yang memanfaatkan teknologi IoT, di mana setiap mobil dipasang *receiver* yang terhubung dengan satu data pengguna, yang bertujuan supaya sistem dapat



mendeteksi posisi kendaraan ketika memasuki gerbang tol tanpa menghambat kendaraan lain. Sistem akan mendeteksi sinyal WiFi terkuat dari gerbang tol, apabila sinyal WiFi terkuat telah sesuai dengan yang terdaftar pada *receiver* akan dibangun koneksi. Setelah koneksi berhasil dibangun, sistem akan mengambil data kendaraan yang telah tersimpan pada basis data dan akan mengubah status kendaraan menjadi “In”, yang berarti telah berhasil memasuki jalan tol. Sistem akan mengulangi proses ini jika sistem mendapati kendaraan dengan status “in”, maka sistem akan kembali mengubah status kendaraan menjadi “out”. Proses transaksi ini dapat dilakukan pada radius 10 meter dari gerbang tol tanpa melakukan pemberhentian dengan kecepatan rata-rata 10 km/jam. Proses transaksi diharapkan dapat dilakukan lebih cepat dengan waktu selama 10 detik untuk gerbang masuk tol dan 15 detik untuk gerbang keluar tol sehingga dapat mengurangi antrian kemacetan yang masih sering terjadi di gerbang pembayaran tol [5].

**c.      **Pemodelan Sistem Pelacakan Lot Parkir Kosong Berbasis Sensor Ultrasonik dan *Internet of things* (IoT) Pada Lahan Parkir Di Luar Jalan [6]****

Kebijakan pemerintah untuk mengurangi adanya parkir di area jalan dilakukan dengan mengharuskan pusat-pusat kegiatan bisnis, perkantoran rumah sakit, tempat wisata dan lainnya untuk menyediakan lahan parkir yang cukup memenuhi standar. Dengan adanya penyediaan lahan parkir yang cukup pada pusat-pusat kegiatan bisnis seperti *mall* diharapkan dapat mengurangi adanya parkir ilegal di jalan. Pada tempat-tempat besar seperti pusat perbelanjaan biasanya memiliki lahan parkir bertingkat. Masalah yang ditimbulkan adalah pencarian tempat parkir yang masih kosong, hal ini menyebabkan kendaraan (mobil) akan berputar-putar atau naik-turun untuk mencari tempat parkir yang masih kosong tersebut. Untuk mengatasi permasalahan tersebut, perlu dilakukan suatu cara agar pengguna lahan parkir dapat dengan mudah mengetahui di mana letak tempat parkir yang masih kosong. Sebuah sensor *ultrasonic* dan IoT dapat digunakan untuk membantu memberikan informasi letak dan status tempat parkir tersebut apakah terisi atau kosong. Alat ini juga berguna bagi manajemen pengelolaan lahan parkir untuk dapat memonitor sirkulasi penggunaan lahan parkir, juga dapat digunakan untuk

monitoring pendapatan. Adapun alat yang digunakan dalam aplikasi yang dibangun adalah sensor ultrasonik dan chip ESP-8266 [6].

## **BAB III**

### **ANALISIS**

Bab ini akan membahas analisis dari aplikasi yang akan dibangun, yang merupakan tahap awal dalam metodologi *scrum*, yaitu *initiate* yang terdiri dari *create project vision, identification scrum master and stakeholders, scrum master, form scrum team, develop epics, content outline, analyze the scrum, create prioritized product backlog* dan *conduct release planning*. Analisis yang akan dilakukan bertujuan untuk menentukan tujuan proyek, ruang lingkup proyek dan melakukan analisis teknologi dan *tools* yang digunakan dalam membangun aplikasi.

#### **3.1 Create Project Vision**

*Create Project Vision* merupakan bagian dari tahap *initiate* yang bertujuan untuk memberikan gambaran singkat tentang aplikasi yang akan dibangun. Pada tahap ini terbagi atas 8 bagian, yaitu latar belakang, tujuan proyek, lingkup proyek, analisis penelitian terkait, persyaratan dan preferensi sistem baru, rencana pengendalian, manajemen resiko dan analisis teknologi.

##### **3.1.1 Latar Belakang**

Latar belakang dapat dilihat pada Bab 1 halaman 1.

##### **3.1.2 Tujuan Proyek**

Tujuan proyek dapat dilihat pada Bab 1 halaman 2.

##### **3.1.3 Lingkup Proyek**

Adapun lingkup proyek dalam tugas akhir ini, yaitu mencari dan mengumpulkan informasi mengenai kebutuhan dalam pembangunan aplikasi melalui internet, referensi jurnal dan buku. Setelah itu, menganalisis, merancang, melakukan implementasi dan melakukan pengujian terhadap aplikasi yang dibangun. Aplikasi yang dibangun akan diterapkan pada kawasan kampus UKDLSM sebagai tempat pengujian aplikasi. Aplikasi ini menggunakan

NodeMCU, sensor ultrasonik dan sensor *accelerometer*, serta algoritma C4.5 untuk melakukan klasifikasi data.

Adapun fitur yang disediakan pada aplikasi ini, yaitu para pengguna dapat melihat informasi keadaan kendaraan di kampus pada aplikasi secara langsung, administrator dapat melihat jumlah data statistik kendaraan yang masuk dan keluar kampus berdasarkan *range* waktu.

### 3.1.4 Analisis Penelitian Terkait

Pada bagian ini akan dilakukan analisis terhadap tiga penelitian terkait. Ketiga penelitian terkait yang telah dibahas pada bab sebelumnya dipilih karena memiliki banyak kesamaan dengan topik tugas akhir yang diambil karena artikel yang dipilih semuanya berbasis IoT dan menggunakan komponen yang serupa dengan komponen yang digunakan dalam pembuatan aplikasi penghitung kendaraan berbasis IoT ini. Untuk memperjelas analisis maka dibuat tabel perbandingan yang dapat di lihat pada Tabel 3.1.

**Tabel 3.1 Analisis Perbandingan Penelitian**

No	Kriteria Perbandingan	Perancangan Sistem Penghitung Jumlah Kendaraan Pada Area Parkir Dengan Metode <i>Background Subtraction</i> Berbasis <i>Internet Of Things</i> [3]	Sistem Deteksi Mobil pada Aplikasi Pembayaran Gerbang Tol Berbasis <i>Internet of Things</i> [5]	Pemodelan Sistem Pelacakan Lot Parkir Kosong Berbasis Sensor Ultrasonik dan <i>Internet of Things</i> Pada Lahan Parkir Di Luar Jalan [6]
1.	Permasalahan	Belum adanya teknologi pada Universitas Diponegoro yang dapat menghitung kendaraan yang	Pesatnya pertumbuhan kendaraan sering menyebabkan kemacetan pada sarana transportasi seperti jalan tol.	Pesatnya pertumbuhan kendaraan menyebabkan sulitnya melakukan pencarian atau pelacakan tempat

No	Kriteria Perbandingan	Perancangan Sistem Penghitung Jumlah Kendaraan Pada Area Parkir Dengan Metode <i>Background Subtraction</i> Berbasis <i>Internet Of Things</i> [3]	Sistem Deteksi Mobil pada Aplikasi Pembayaran Gerbang Tol Berbasis <i>Internet of Things</i> [5]	Pemodelan Sistem Pelacakan Lot Parkir Kosong Berbasis Sensor Ultrasonik dan <i>Internet of Things</i> (IoT) Pada Lahan Parkir Di Luar Jalan [6]
		masuk dan keluar secara <i>live</i> .		parkir yang masih kosong. Kendaraan kesulitan untuk putar balik untuk mencari tempat parkir kosong.
2.	Tujuan	Untuk menganalisis jumlah kendaraan yang keluar masuk pada area parkir terbuka di sekitar kampus.	Aplikasi ini diharapkan dapat mempercepat proses transaksi pada tol.	Untuk mengetahui letak tempat parkir yang masih kosong.
3.	Peralatan	Kamera.	NodeMCU, buzzer, LCD.	Sensor ultrasonik, NodeMCU, RFID
4.	Objek	Mobil dan motor.	Mobil.	Mobil.
5.	Pengujian	Masukan yang digunakan adalah video dengan durasi berbeda yang direkam dengan laju <i>frame</i> sebanyak 24	Pengujian dilakukan pada 2 sisi gerbang tol dengan jarak 10-20 meter. Percobaan	Alat dipasang pada plafon di atas tempat parkir dengan posisi tengah-tengah pada lokasi parkir

No	Kriteria Perbandingan	Perancangan Sistem Penghitung Jumlah Kendaraan Pada Area Parkir Dengan Metode <i>Background Subtraction</i> Berbasis <i>Internet Of Things</i> [3]	Sistem Deteksi Mobil pada Aplikasi Pembayaran Gerbang Tol Berbasis <i>Internet of Things</i> [5]	Pemodelan Sistem Pelacakan Lot Parkir Kosong Berbasis Sensor Ultrasonik dan <i>Internet of Things</i> (IoT) Pada Lahan Parkir Di Luar Jalan [6]
		<p><i>frame</i> per detik. Posisi kamera tidak berubah pada berbagai pengujian. Seluruh pengujian dilakukan pada pagi pukul 08.00 sampai pukul 21.00.</p>	<p>digunakan dengan menggunakan mobil yang telah terpasang <i>receiver</i>, serta keadaan kaca mobil terbuka sebanyak 30 kali.</p>	<p>yang telah ditentukan sebagai tempat uji lapangan. Setelah mendeteksi objek maka data akan dikirim ke server untuk diolah.</p>
6.	Akurasi	<p>Akurasi tingkat keberhasilan pada pengujian data per hari adalah 90,57%, sedangkan pada pengujian per jam akurasi tertinggi terdapat pada jam 08.00 – 10.00 yaitu 100% untuk mobil dan 96% untuk motor.</p>	<p>Dari 30 kali percobaan, terdapat 3 kali percobaan yang gagal yaitu pada kondisi kecepatan 20 Km/h dan jarak 10 sampai 15 meter. Sedangkan 27 percobaan lainnya sukses.</p>	-

Adapun kelebihan dan kekurangan ketiga penelitian pada Tabel 3.1, yaitu: Pada Perancangan Sistem Penghitung Jumlah Kendaraan Pada Area Parkir Dengan Metode *Background Subtraction* Berbasis *Internet of Things* [3] menggunakan sistem pengolahan citra dengan metode *background subtraction* di mana data yang dihasilkan berupa gambar digital yang digunakan sebagai pengklasifikasi dan pendeteksi objek yang lewat. Hal ini membuat sistem yang dibangun memiliki kelebihan, yaitu akurasi yang relatif tinggi dalam mengklasifikasikan objek yang lewat yaitu 90,57% untuk pengujian perhari, namun sistem ini hanya mendeteksi objek motor dan mobil saja. Faktor cuaca juga mempengaruhi pengambilan gambar, kemudian daya yang diperlukan untuk menjalankan sistem ini terbilang besar jika dibandingkan dengan penelitian yang lain. Selain itu, harga peralatannya relatif lebih mahal. Selanjutnya, Sistem Deteksi Mobil pada Aplikasi Pembayaran Gerbang Tol Berbasis *Internet of Things* [5] menggunakan RFID (*Radio Frequency Identification*) untuk mendeteksi kendaraan yang lewat. Kelebihan sistem pada penelitian ini adalah proses transaksi sangat praktis dan cepat dikarenakan penggunaan RFID yang memungkinkan pengendara melakukan transaksi dengan hanya melewati sensor yang terpasang pada pintu tol. Sistem ini memiliki kekurangan yaitu kendaraan yang lewat harus mengurangi kecepatannya sampai dengan 10 km/jam dari radius 10 meter sebelum sensor, objek yang dideteksi hanya untuk mobil. Selain itu, setiap pengendara harus mempunyai alat pendeteksi RFID dan harus dalam keadaan aktif ketika akan melewati gerbang tol. Pemodelan Sistem Pelacakan Lot Parkir Kosong Berbasis Sensor Ultrasonik dan *Internet of Things* (IoT) Pada Lahan Parkir Di Luar Jalan [6] menggunakan sensor ultrasonik untuk mendeteksi objek di setiap tempat parkir. Keuntungan dari sistem ini adalah biaya peralatan yang lebih murah jika dibandingkan dengan kedua penelitian sebelumnya. Namun sistem ini memiliki kelemahan, yaitu dikarenakan kendaraan memiliki bentuk permukaan yang berbeda-beda, jika posisi permukaan kendaraan tidak sesuai dengan sensor ultrasonik yang dipasang atau permukaan kendaraan menggunakan material bahan yang tidak memantulkan gelombang, maka sensor

ultrasonik tidak dapat mendeteksi kendaraan dengan maksimal. Selain itu objek yang dideteksi tidak dibatasi.

### 3.1.5 Persyaratan dan Preferensi Sistem Baru

Pada tahap persyaratan dan preferensi sistem baru ini akan dilakukan analisis terhadap aplikasi yang dibangun. Pada tahap sebelumnya telah dijelaskan kelebihan dan kekurangan dari penelitian terkait. Aplikasi yang dibangun ini memiliki beberapa kelebihan dan fitur-fitur yang ditambahkan. Kelebihan aplikasi yang dibangun ini, yaitu memiliki biaya peralatan yang lebih murah, menggunakan 2 sensor, yaitu sensor *accelerometer* untuk mendeteksi getaran objek dan sensor ultrasonik yang digunakan untuk mendeteksi objek yang lewat, menggunakan algoritma C4.5 yang akan membantu pengklasifikasian objek lebih akurat. Adapun fitur-fitur pada aplikasi ini, antara lain:

- a. Menghitung jumlah kendaraan yang masuk dan keluar secara *real time*.
- b. Memberikan informasi status tempat parkir di UKDLSM.
- c. Membuat laporan keluar masuk kendaraan.

### 3.1.6 Rencana Pengendalian

Bagian ini akan menguraikan rencana pengendalian dari pembangunan aplikasi ini yang dapat dilihat pada Tabel 3.2.

**Tabel 3.2 Tabel Rencana Pengendalian**

Rencana Awal	Perubahan Rencana	Rencana Akhir	Alasan Perubahan
1. Sensor yang digunakan berjumlah 3 buah, yaitu sensor <i>accelerometer</i> , sensor ultrasonik dan sensor LDR	1. Sensor yang digunakan hanya 2 buah, yaitu sensor <i>accelerometer</i> dan sensor ultrasonik. Sensor LDR	1. Sensor yang digunakan hanya 2 buah, yaitu sensor <i>accelerometer</i> dan sensor ultrasonik.	1. Sensor ultrasonik dan sensor LDR memiliki fungsi yang hampir sama.



Rencana Awal	Perubahan Rencana	Rencana Akhir	Alasan Perubahan
<p>(<i>Light Dependent Resistor</i>).</p> <p>2. Metodologi pengembangan sistem yang akan digunakan adalah metodologi RUP (<i>Rational Unified Process</i>).</p> <p>3. Menggunakan Arduino Uno.</p>	<p>tidak lagi digunakan.</p> <p>2. Metodologi pengembangan sistem yang sebelumnya RUP diganti menjadi metodologi Scrum.</p> <p>3. Tidak lagi menggunakan Arduino Uno.</p>	<p>2. Metodologi pengembangan sistem yang digunakan yaitu Scrum.</p> <p>3. Hanya menggunakan NodeMCU.</p> <p>4. Fitur melihat laporan detail informasi kendaraan keluar/masuk ditambahkan.</p>	<p>2. Metodologi pengembangan sistem Scrum lebih cocok dengan IoT.</p> <p>3. NodeMCU sudah cukup untuk menangani 2 sensor dan pengiriman data.</p> <p>4. Untuk menambah fitur aplikasi.</p>

### 3.1.7 Manajemen Resiko

Pada tahap manajemen resiko ini akan dilakukan analisis terhadap resiko yang terjadi pada tahap pembuatan aplikasi penghitung kendaraan berbasis IoT yang dapat dilihat pada Tabel 3.3 berikut ini.

**Tabel 3.3 Manajemen Resiko**

No	Resiko	Penyebab	Efek
1.	Kerusakan perangkat yang digunakan selama pembuatan aplikasi.	<i>Human error.</i>	Tidak dapat menyelesaikan aplikasi sesuai dengan jadwal.
2.	Perubahan jadwal.	Wabah covid-19.	Waktu yang dibutuhkan untuk menyelesaikan aplikasi lebih lama.
3.	Kesalahan pemilihan sensor yang akan digunakan.	Mahasiswa tidak memahami sensor yang akan digunakan.	Perubahan pada sensor yang akan digunakan.

### 3.1.8 Analisis Teknologi

Di bawah ini merupakan analisis teknologi yang digunakan dalam pembuatan aplikasi dalam hal ini yaitu *software* dan *hardware*. Berikut ini daftar *software* dan *hardware* yang digunakan.

**Tabel 3.4 Daftar Teknologi yang Digunakan**

No	Perangkat Lunak ( <i>software</i> )	
1.	Pemrograman	JavaScript dan PHP versi 7
2.	DBMS	MySQL 5.0.11
3.	Web Server	Apache 2.4.23
4.	Browser	Firefox 70.0.1 (64-bit)
5.	Sistem Operasi	Windows 10 Pro
6.	Text Editor	Visual Studio Code versi 1.38.1

7.	Penggambaran kaskas pemodelan	<i>Lucid Chart Online, Edraw 7.9</i>
<b>No</b>	<b>Perangkat Keras (<i>hardware</i>)</b>	
1.	<i>Processor</i>	Intel Core i5-8250U
2.	<i>Memory</i>	8GB DDR4
4.	Peralatan Pendukung	<i>Mouse dan keyboard.</i>
5.	Sensor	Sensor <i>accelerometer</i> dan sensor <i>ultrasonic.</i>

### 3.2 **Identification Scrum Master and Stakeholders**

Pada tahap *identification scrum master and stakeholders* ini merupakan bagian dari tahap *initiate* dari metodologi *scrum* yang terdiri dari *scrum master*, analisis pengguna, spesifikasi kebutuhan dan usulan solusi yang akan dijelaskan sebagai berikut.

#### 3.2.1 **Scrum Master**

*Scrum master* merupakan seseorang yang bertanggung jawab dalam memimpin dan membimbing selama proses pembuatan aplikasi. Dalam tugas akhir ini yang menjadi *scrum master* yaitu Lanny Sitanayah, S.T., M.Sc., Ph.D. selaku *supervisor* I dan Apriandy Angdresey, S.T., M.Sc. selaku *supervisor* II.

#### 3.2.2 **Analisis Pengguna**

Pada tahap ini akan menjelaskan tentang analisis pengguna dari aplikasi yang dibangun, manfaat yang akan didapatkan pengguna dari aplikasi. Pengguna aplikasi ini adalah pegawai yang bertugas mengawasi tempat parkir di Universitas Katolik De La Salle Manado dan para pengendara yang hendak memarkirkan kendaraannya Universitas Katolik De La Salle. Untuk lebih jelasnya dapat dilihat pada tabel di bawah ini.

**Tabel 3.5 Tabel Analisis Pengguna**

<b>Identifikasi Pengguna</b>	<b>Manfaat</b>
Pengguna	1. Menghitung jumlah kendaraan yang masuk dan keluar.

Identifikasi Pengguna	Manfaat
	2. Memberikan laporan tentang informasi status tempat parkir di UKDLSM.

### 3.2.3 Spesifikasi Kebutuhan

Pada tahap ini merupakan penjelasan dari spesifikasi kebutuhan yang bertujuan untuk menjelaskan fungsi-fungsi yang akan didapatkan pengguna dari aplikasi yang dibangun secara keseluruhan berdasarkan fungsional dan non-fungsional.

**Tabel 3.6 Tabel Spesifikasi Kebutuhan**

Fungsional	Non fungsional
1. Menghitung jumlah kendaraan yang masuk dan keluar secara. 2. Memberikan informasi status tempat parkir di UKDLSM. 3. Melihat laporan jumlah kendaraan keluar masuk.	1. Aplikasi memiliki tampilan yang sederhana dengan beberapa tampilan informasi dari jumlah kendaraan di UKDLSM dan informasi status lahan parkir. 2. Aplikasi dapat dijalankan pada <i>web browser</i> yang umum digunakan pada masyarakat.

### 3.2.4 Usulan Solusi

Pada bagian usulan solusi ini akan dilakukan analisis terhadap aplikasi penghitung kendaraan berbasis IoT. Aplikasi yang dibangun ini memiliki beberapa proses alur, yang pertama aplikasi akan mengumpulkan data dari sensor, yaitu sensor *accelerometer* dan ultrasonik. Data tersebut akan diolah menjadi data *training* untuk diklasifikasi menggunakan algoritma C4.5. Setelah selesai, data akan di-*input* ke basis data, lalu aplikasi akan melakukan perhitungan jumlah kendaraan. Untuk penjelasan lebih lanjut tentang pengklasifikasian algoritma C4.5 ini, akan dilakukan pengujian terhadap data yang telah dikumpulkan sebelumnya yang dapat dilihat pada perhitungan di bawah ini. Pada tabel ini memiliki enam atribut yaitu sensor ultrasonik, sensor *accelerometer* AX, AY, AZ, *times* dan kelas.

Atribut sensor ultrasonik menampung data dalam satuan sentimeter. Pada sensor *accelerometer* AX, AY, AZ merupakan singkatan dari *accelerometer* pada sumbu X, Y, Z. Atribut *times* merupakan banyaknya data yang ditangkap dalam satu objek, sedangkan atribut kelas merupakan kelas objek yang akan diklasifikasikan yang terbagi menjadi tiga, yaitu mobil, motor dan orang.

**Tabel 3.7 Data Training**

Sensor ultrasonik	Sensor <i>accelerometer</i> AX	Sensor <i>accelerometer</i> AY	Sensor <i>accelerometer</i> AZ	<i>times</i>	kelas
82,00	-1,73	3,43	3,12	2	orang
81,33	-1,43	3,33	3,18	3	mobil
72,33	-1,60	3,47	3,05	3	mobil
94,00	-1,40	3,44	3,06	1	orang
84,67	-1,67	3,42	3,12	3	motor
87,50	-1,67	3,34	3,06	2	motor
80,00	-1,60	3,47	3,07	5	mobil
87,67	-1,70	3,25	3,13	3	motor
70,23	-1,49	3,22	3,06	1	orang

Pada Tabel 3.8 data telah diubah dalam bentuk diskrit yaitu rendah, sedang, tinggi yang bertujuan mempermudah perhitungan algoritma. Bentuk diskrit diperoleh dari perhitungan sebagai berikut:

$$Range = X_{max} - X_{min}$$

$$Range(S) = \frac{Range}{3}$$

$$Kategori\ 1 = X_{min} + Range(S)$$

$$Kategori\ 2 = kategori\ 1 + Range(S)$$

$$Kategori\ 3 = kategori\ 2 + Range(S)$$

Contoh perhitungan mencari *range* nilai kategori pada kolom sensor ultrasonik pada Tabel 3.8:

$$Range = 94,00 - 70,23 = 23,77$$

$$Range(S) = \frac{23,77}{3} = 7,923$$

∴

3

$$Kategori\ 1 = 70,23 + 7,923 = 78,153$$

Kategori 2 =  $78,153 + 7,923 = 78,153$

Kategori 3 =  $86,076 + 7,923 = 86,076$

Jadi, *range* nilai kategori dari kolom sensor ultrasonik sebagai berikut:

$\geq 70,23$  dan  $< 78,153$  = dekat

$\geq 78,153$  dan  $< 86,076$  = menengah

$\geq 86,076$  dan  $< 94,00$  = jauh

**Tabel 3.8 Konversi bentuk Diskrit**

Sensor ultrasonik	Sensor <i>accelerometer</i> AX	Sensor <i>accelerometer</i> AY	Sensor <i>accelerometer</i> AZ	<i>times</i>	kelas
menengah	rendah	sedang	sedang	rendah	orang
menengah	tinggi	sedang	tinggi	sedang	mobil
dekat	sedang	tinggi	tinggi	tinggi	mobil
jauh	rendah	sedang	rendah	rendah	orang
menengah	rendah	tinggi	sedang	sedang	motor
jauh	sedang	sedang	rendah	rendah	motor
menengah	sedang	tinggi	tinggi	tinggi	mobil
jauh	sedang	rendah	tinggi	sedang	motor
dekat	sedang	rendah	rendah	rendah	orang

**Tabel 3.9 Mencari Entropy dan Gain**

		Jumlah (S)	Orang (Si)	Motor (Si)	Mobil (Si)	Entropy	Gain
total		9	3	3	3	1,584962501	
us							
	dekat	2	1	0		0	0,91829583
	menengah	4	1	1	2	1,5	
	jauh	3	1	2	0	0	
ax							
	rendah	3	2	1	0	0	0,91829583
	sedang	4	1	2	2	1,5	

		Jumlah (S)	Orang (Si)	Motor (Si)	Mobil (Si)	Entropy	Gain
	tinggi	2	0	0	1	0	
ay							
	rendah	2	1	1	0	0	0,91829583
	sedang	4	2	1	2	1,5	
	tinggi	3	0	1	1	0	
az							
	rendah	3	2	1	0	0	1,36274028
	sedang	2	1	1	2	1	
	tinggi	4	0	1	1	0	
waktu		9	3	3	3	1,584962501	
	rendah	4	3	1	0	0	1,5849625
	sedang	3	0	2	1	0	
	tinggi	2	0	0	2	0	

Pada Tabel 3.9 dilakukan perhitungan untuk mencari nilai entropy dan gain yang menggunakan formula yang dibahas pada bab sebelumnya.

#### Entropy total (S)

$$= ((-3/9) * \log_2(3/9)) + ((-3/9) * \log_2(3/9)) + ((-3/9) * \log_2(3/9))$$

$$= 1,584962501$$

#### Gain (S,A)

$$= (1,584962501) - ((2/9) * 0) - ((4/9) * 1,5) - ((3/9) * 0)$$

$$= 0,918295834$$

### 3.2.5 Form Scrum Team

Di bawah ini merupakan struktur organisasi dari aplikasi yang dibangun, dapat dilihat pada Tabel 3.10.

**Tabel 3.10 Scrum Team**

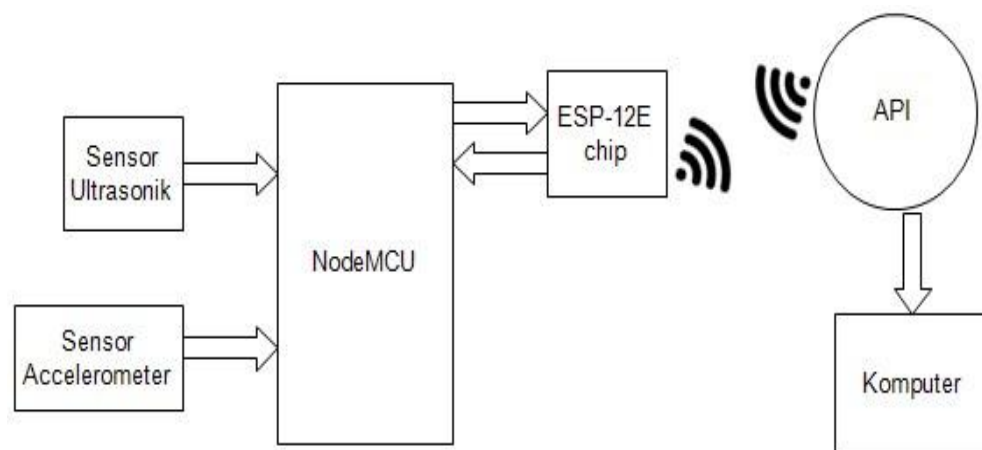
Nama	Peran
Lanny Sitanayah, S.T., M.Sc., Ph.D.	Supervisor I (Scrum Master)
Apriandy Angdresey, S.T., M.Sc.	Supervisor II (Scrum Master)

Nama	Peran
Jeri Wahyu utama	<i>System Analyst, Designer dan Programmer.</i>

### 3.3 Develop Epics

*Develop Epics* merupakan bagian dari tahap *initiate* pada metodologi *scrum*. Tahap ini akan menjelaskan tentang *content outline* dan *analyze the scrum*. Untuk mendeskripsikan alur dari aplikasi yang dibangun maka digunakan *flowchart*.

#### 3.3.1 Content Outline



Berikut ini merupakan gambaran *content outline* dari aplikasi penghitung kendaraan berbasis IoT, yang ditampilkan pada Gambar 3.1 di halaman selanjutnya. Desain sistem yang memiliki alur, yaitu data dari sensor dikirim melalui NodeMCU yang kemudian diterima oleh API dan diteruskan ke aplikasi.

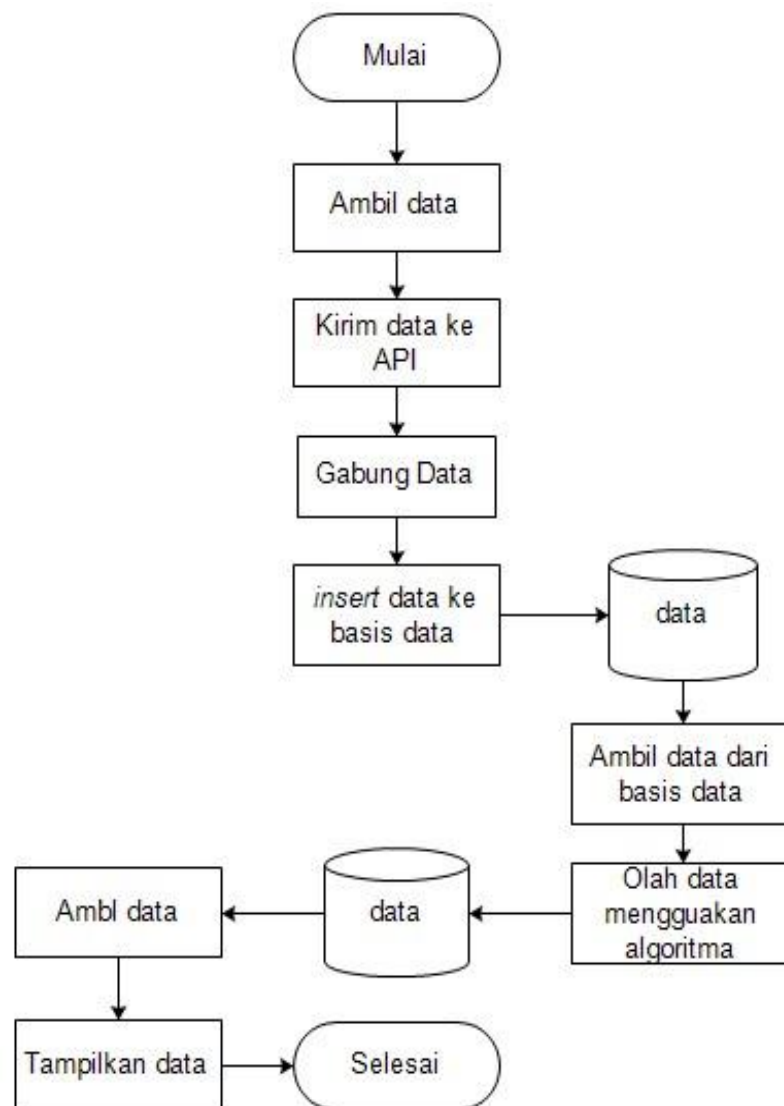
**Gambar 3.1 Content Outline Aplikasi Penghitung Kendaraan Berbasis IoT**

#### 3.3.2 Analyze the Scrum

Aplikasi yang dibangun ini memiliki sisi *client* dan *server*. Sisi *client* ini termasuk komponen *hardware* IoT dan *script* arduino yang berfungsi untuk memperoleh data dan mengirim ke sisi *server*. Sisi *server* ini berfungsi untuk menerima data yang dikirimkan, memproses data dan menampilkan data. Sisi *server* di sini adalah API sebagai *back end* yang menerima dan memproses data



dan aplikasi sebagai *front end* yang bertugas untuk menampilkan dan beberapa proses lain. Aplikasi memiliki beberapa proses yaitu pengambilan data menggunakan sensor-sensor yang kemudian dikirimkan ke REST API, setelah itu *server* akan mengirimkan data tersebut ke komputer. Selanjutnya akan dilakukan penerapan algoritma C4.5 untuk melakukan klasifikasi objek, dan terakhir dilakukan perhitungan jumlah kendaraan yang keluar dan masuk sehingga aplikasi dapat memberikan informasi ketersediaan lahan parkir. Berikut ini merupakan penjelasan tentang alur proses pada aplikasi penghitung kendaraan berbasis IoT yang digambarkan dalam bentuk diagram alir, yang dapat dilihat pada Gambar 3.2 berikut.



**Gambar 3.2 Flowchart Aplikasi Penghitung Kendaraan Berbasis IoT**

### 3.4 Create Prioritized Product Backlog

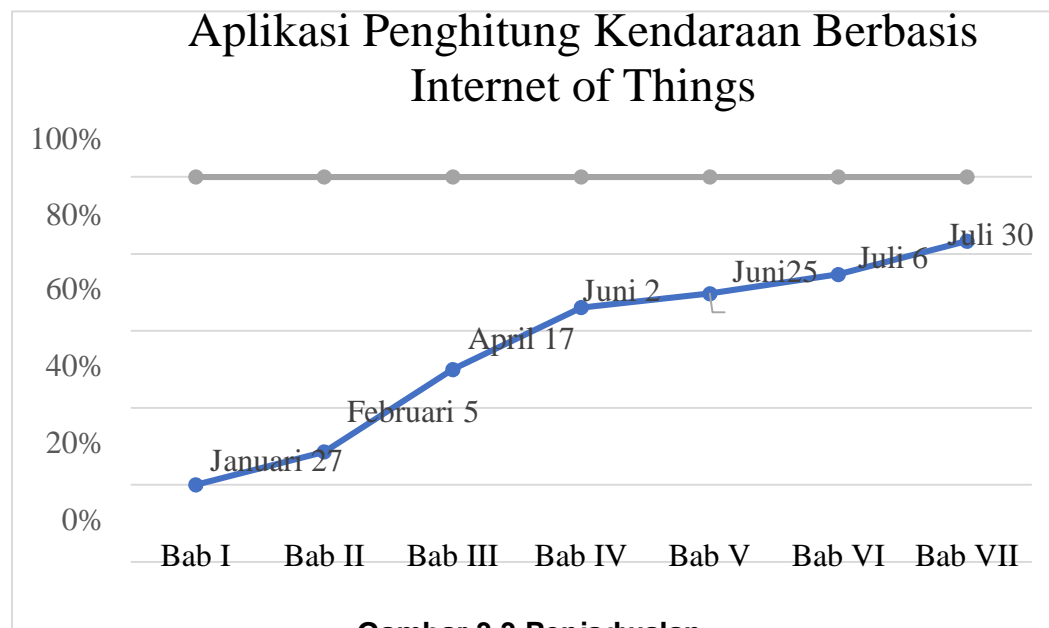
Pada bagian ini ada beberapa fitur utama pada aplikasi yang dibangun yang dapat dilihat pada Tabel 3.11.

**Tabel 3.11 Tabel Fitur**

Nama Fitur	Deskripsi	Prioritas
Menghitung jumlah kendaraan	Menghitung jumlah kendaraan yang masuk dan keluar secara.	Tinggi
Memberikan informasi	Memberikan informasi status tempat parkir di UKDLSM.	Tinggi
Melihat laporan	Melihat laporan keluar masuk kendaraan.	Tinggi

### 3.5 Conduct Release Planning

Berikut merupakan penjadwalan dari pengembangan aplikasi penghitung kendaraan berbasis IoT.



**Gambar 3.3 Penjadwalan**

## BAB IV

### PERANCANGAN

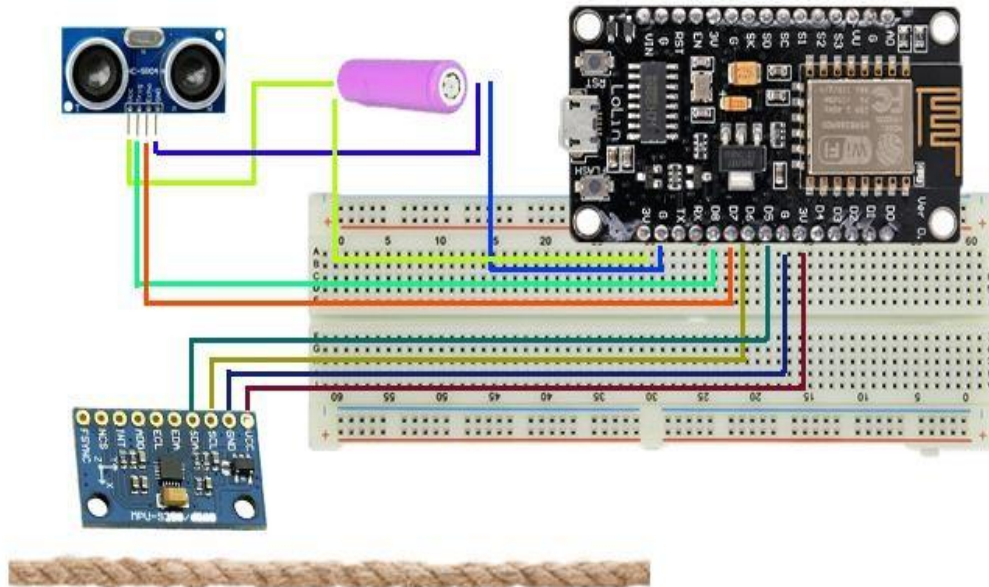
Pada bab ini akan membahas mengenai perancangan aplikasi berdasarkan hasil analisis yang sudah dilakukan pada bab sebelumnya. Perancangan yang dilakukan merupakan bagian dari tahap kedua dari metodologi *scrum*, yaitu *plain and estimate* yang terdiri dari *create user story*, *estimate task* dan *create sprint backlog*

#### 4.1 **Create User Story**

Pada tahap ini akan membahas tentang rancangan *hardware*, rancangan aplikasi dan rancangan basis data dari aplikasi yang dibuat.

##### 4.1.1 **Rancangan Hardware**

Pada tahap ini menggambarkan rancangan dari *hardware* yang akan dibangun. *Hardware* ini dirancang menggunakan sensor ultrasonik dan NodeMCU yang terpasang pada *breadboard*, dan sensor *accelerometer* yang terpasang pada tali unuk menjadi media penghantar getar sehingga memaksimalkan potensi alat. Sensor ultrasonik dan NodeMCU menggunakan daya dari baterai *eksternal*, sedangkan *accelerometer* menggunakan daya dari NodeMCU. Rancangan rangkaian alat ini dapat dilihat pada Gambar 4.1, di mana terdapat NodeMCU, sensor *accelerometer*, sensor ultrasonik, *Breadboard* dan baterai 16650 yang saling terhubung dengan kabel. Pin D5 pada NodeMCU, terhubung ke pin SDA dari sensor *accelerometer*. Pin D6 terhubung ke pin SCL pada sensor *accelerometer*, sedangkan pin GND pada NodeMCU terhubung ke pin GND pada sensor *accelerometer*. Pin 3V pada NodeMCU Terhubung ke pin VCC pada sensor *accelerometer* dan pin D7 pada NodeMCU terhubung ke pin *echo* pada sensor ultrasonik. Pin D8 NodeMCU terhubung ke pin *trigger* pada sensor ultrasonik.

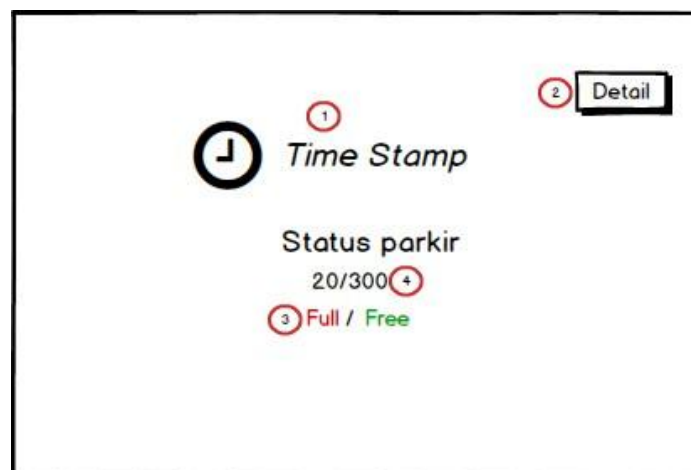


**Gambar 4.1 Rancangan *Hardware***

#### 4.1.2 Rancangan Aplikasi

Pada tahap ini akan menggambarkan desain dari aplikasi penghitung kendaraan berbasis IoT yang akan dibangun yaitu desain antarmuka (*storyboard*).

##### 1). Halaman Utama

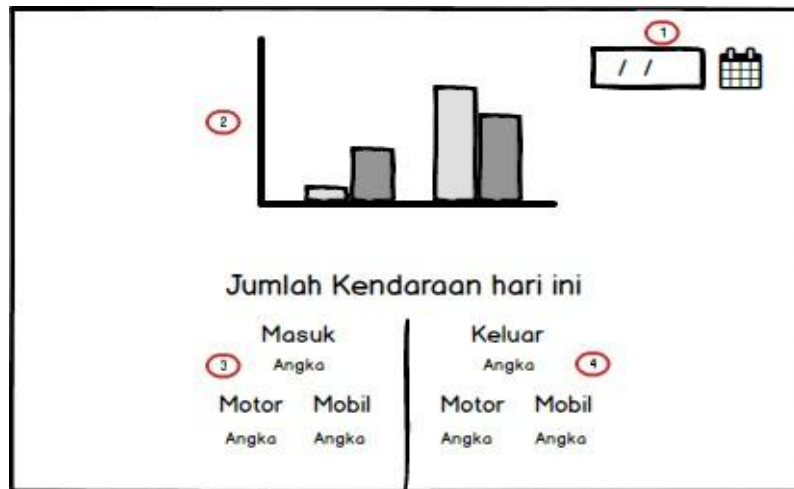


**Gambar 4.2 *Storyboard* Halaman Utama**

Keterangan Gambar 4.2:

1. Jam digital yang bertujuan untuk menunjukkan waktu pada saat ini.
2. Tombol bertujuan untuk masuk ke dalam halaman detail.
3. Status parkir pada saat ini. Status parkir memiliki 2 nilai, yaitu *full* dan *free*.
4. Jumlah kendaraan yang masuk.

## 2). Halaman Detail



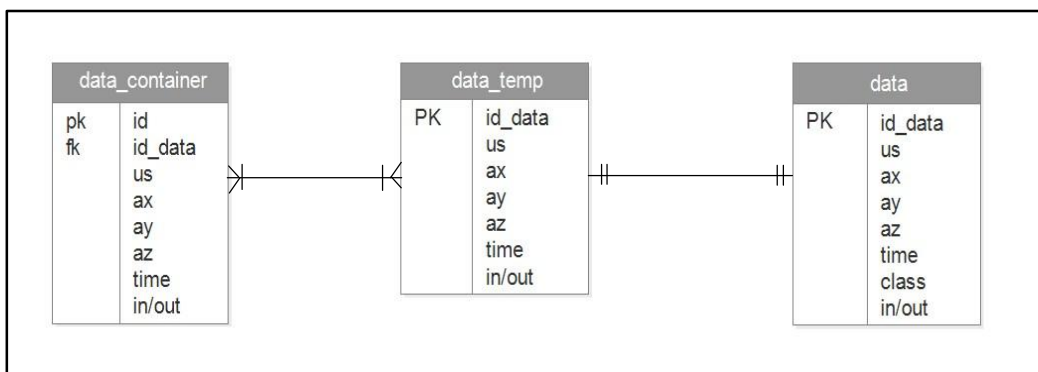
Gambar 4.3 Storyboard Halaman Detail

Keterangan Gambar 4.3:

1. *Date chooser* yang digunakan untuk mengubah data *chart* sesuai tanggal yang dipilih.
2. *Chart* untuk menunjukkan data kendaraan yang masuk keluar berdasarkan waktu yang dipilih.
3. Keterangan jenis kendaraan dan jumlah kendaraan yang masuk pada waktu sekarang.
4. Keterangan jenis kendaraan dan jumlah kendaraan yang keluar pada waktu sekarang.

#### 4.1.3 Rancangan Basis Data

Tahap ini merupakan rancangan basis data yang dibuat untuk aplikasi yang bertujuan untuk menggambarkan relasi antar objek pada basis data. Penggambaran basis data ini menggunakan ERD yang dapat dilihat pada Gambar 4.4.



Gambar 4.4 ERD Aplikasi Penghitung Kendaraan Berbasis IoT

## 4.2 Estimate Task

Berikut merupakan estimasi waktu pengerjaan dari aplikasi yang dibangun yang disajikan dalam bentuk tabel.

**Tabel 4.1 Estimate Task**

<i>Task</i>	<i>Estimate</i>
Halaman Utama	14 Hari
Halaman Admin	14 Hari
Perancangan Alat	1 Bulan
Program Alat	1 Bulan
Program Algoritma	1 Bulan
Fitur-fitur lainnya	14 Hari

## 4.3 Rancangan Modul Program

Berikut merupakan rancangan modul program yang ada pada aplikasi penghitung kendaraan berbasis IoT.

**Tabel 4.2 Rancangan Modul Program**

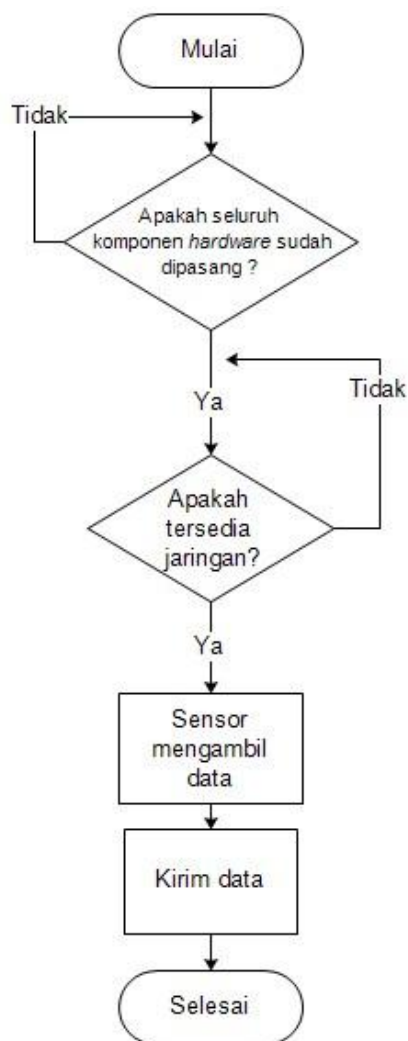
No	Modul	Fungsi
1.	Fungsi <i>data-grouping</i>	Fungsi ini bertujuan untuk mengubah bentuk data yang awalnya numerik menjadi data diskrit, agar lebih mudah diolah oleh algoritma, khususnya C4.5.
2.	Fungsi <i>POST</i>	Fungsi ini digunakan untuk mengirim data dari sensor ke API ( <i>Application Programming Interface</i> ).
3.	Fungsi Perhitungan Algoritma	Fungsi ini merupakan fungsi utama dalam aplikasi. Fungsi ini digunakan untuk menghitung nilai <i>entropy</i> , nilai <i>gain</i> dan lain sebagainya.
4.	Koneksi basis data	Fungsi ini digunakan untuk menghubungkan aplikasi yang dibangun dengan basis data.

#### 4.4 Create Sprint Backlog

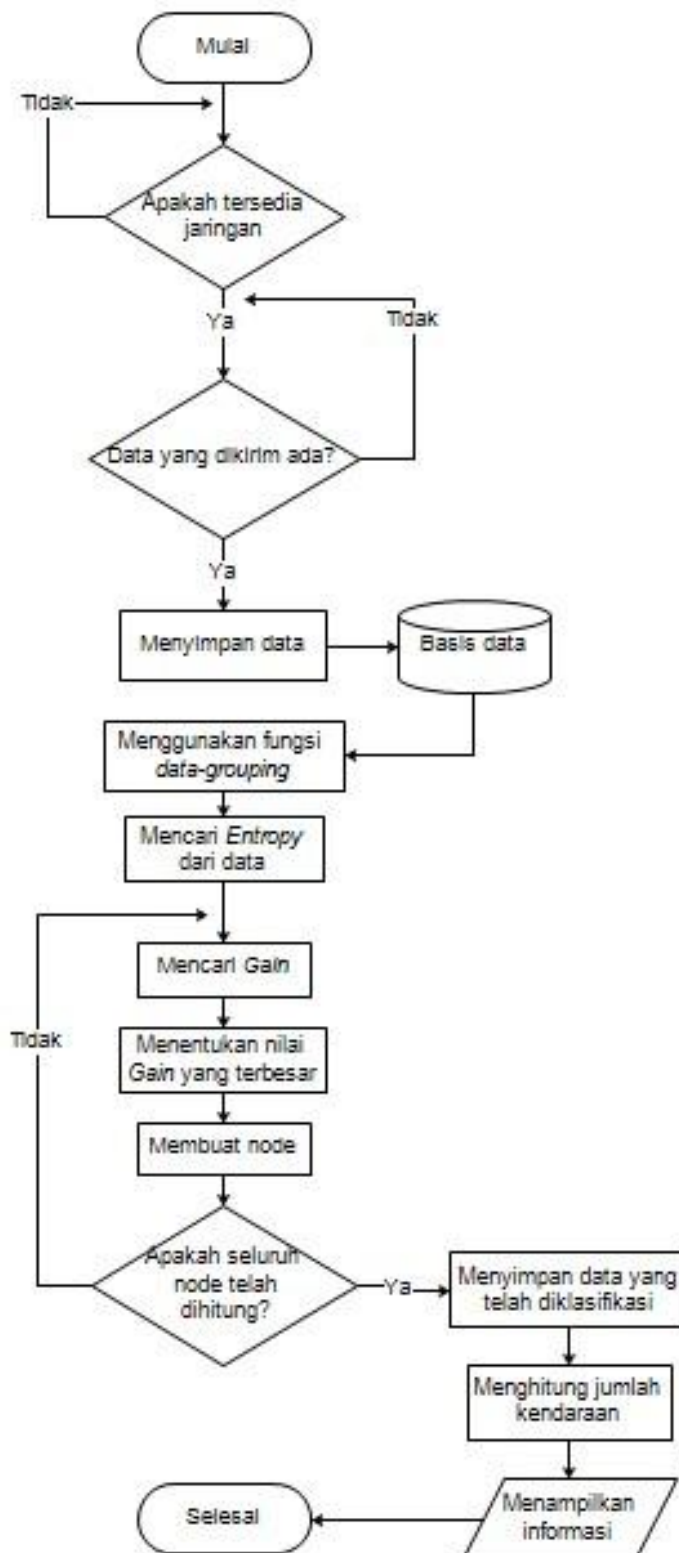
Pada tahap ini akan menggambarkan urutan proses dari aplikasi yang dibangun. Diagram yang digunakan adalah *flowchart* untuk menampilkan setiap alur pada aplikasi yang dibuat.

##### 4.4.1 Flowchart

*Flowchart* ini menunjukkan alur kerja pada aplikasi penghitung kendaraan berbasis IoT yang terbagi menjadi dua bagian yaitu dari sisi *client* dan *server*, yang ditunjukkan pada Gambar 4.5 dan Gambar 4.6.



Gambar 4.5 Flowchart Client



**Gambar 4.6 Flowchart Server**



## BAB V IMPLEMENTASI

Bab ini akan membahas tentang implementasi yang merupakan bagian dari tahap ketiga metodologi *scrum*, yaitu *implement*. Bab ini merupakan implementasi dari bab sebelumnya yang dilakukan dengan menggunakan *JavaScript* dan *PHP*.

### 5.1 **Create Deliverable**

*Create deliverable* ini akan menjelaskan tentang tugas-tugas yang ada pada *sprint backlog* yang bertujuan untuk mengetahui kegiatan dan pekerjaan serta permasalahan yang terjadi selama implementasi program.

#### 5.1.1 **Lingkungan Implementasi**

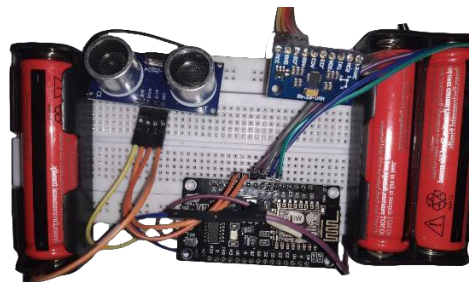
Tahap ini akan membahas tentang spesifikasi *software* dan *hardware* yang digunakan dalam menjalankan aplikasi ini yang dapat dilihat pada Tabel 5.1.

**Tabel 5.1 Spesifikasi Software dan Hardware**

No	Perangkat Lunak ( <i>software</i> )	
1.	<i>Browser</i>	<i>Web browser</i> yang umum digunakan.
2.	Sistem Operasi	Windows, Ios, Android, Linux
No	Perangkat Keras ( <i>hardware</i> )	
1.	<i>Processor</i>	Intel Core i5-8250U
2.	<i>Memory</i>	Minimal 2GB DDR4

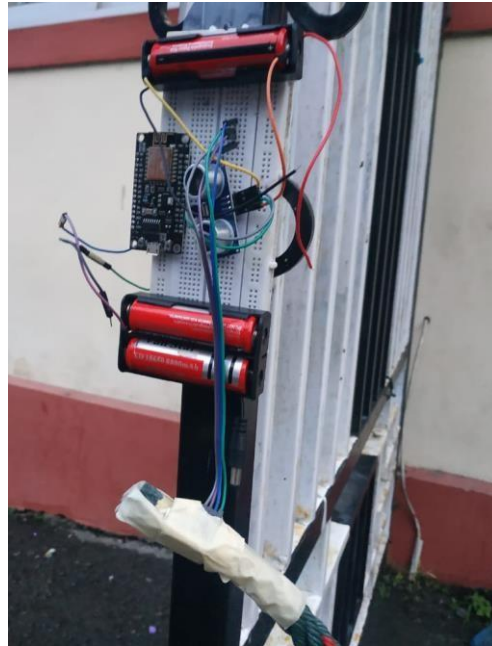
#### 5.1.2 **Implementasi Hardware**

Hasil dari implementasi alat yang telah berhasil dibangun sesuai dengan perancangan pada bab sebelumnya ditampilkan pada Gambar 5.1.



**Gambar 5.1 Hardware Pendeteksi Kendaraan**

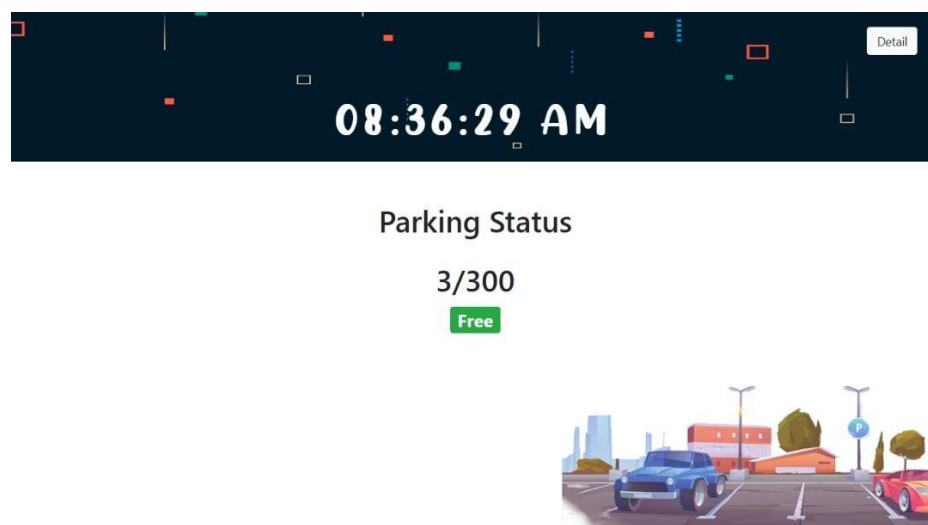
Selanjutnya implementasi alat yang dipasang pada portal, dengan ketinggian 0,25 meter dari permukaan tanah. Implementasi alat ini juga menggunakan seutas tali yang dipasangkan dengan sensor *accelerometer* yang dibiarkan membujur di permukaan tanah sehingga sensor dapat mendeteksi setiap objek yang melewatinya, dapat dilihat pada halaman berikutnya Gambar 5.2.



Gambar 5.2 Implementasi *Hardware* Pada Portal

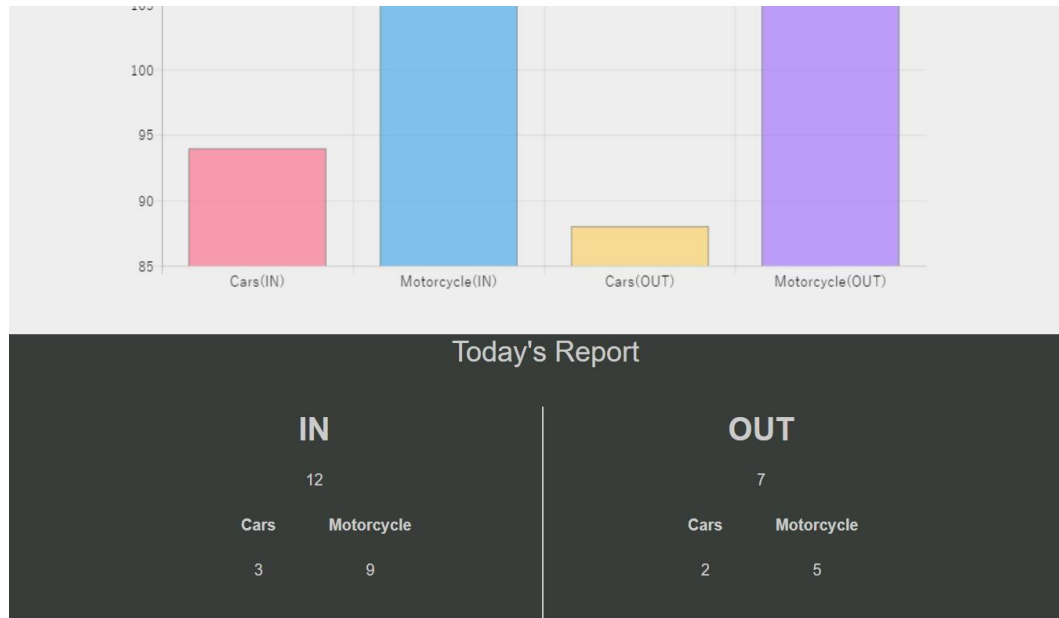
### 5.1.3 Implementasi *Software*

Tahap ini berisi tentang tampilan antarmuka aplikasi yang telah dirancang pada bab sebelumnya.



Gambar 5.3 Halaman Utama

Gambar 5.3 merupakan halaman utama dari aplikasi yang dibangun. Pada halaman ini terdapat jam digital pada bagian atas aplikasi dan status parkir pada bagian tengah, yang menunjukkan status parkir.



**Gambar 5.4 Halaman Detail**

Gambar 5.4 merupakan implementasi dari *storyboard* yang dibuat sebelumnya. Halaman detail ini berisi informasi detail dari data-data yang diperoleh. Pada halaman ini terdapat *bar chart* yang bertujuan menginformasikan berapa banyak mobil dan motor yang masuk dan keluar sesuai dengan waktu yang dimasukkan. Kemudian pada bagian bawah terdapat informasi tentang jumlah kendaraan yang masuk dan keluar pada hari aplikasi sedang dioperasikan.

#### 5.1.4 Implementasi Basis Data

Tahap ini berisikan tampilan basis data yang diimplementasikan ke dalam MySQL. Gambar 5.5 merupakan tabel basis data secara keseluruhan, kemudian Gambar 5.6 merupakan tabel data container, yang mana tabel ini digunakan untuk menampung data mentah dari sensor. Gambar 5.7 merupakan tabel data temp, yang digunakan untuk sebagai tempat menaruh data yang telah diproses menggunakan fungsi *data-grouping*. Gambar 5.8 merupakan tabel data yang berisikan data yang sudah selesai melewati proses algoritma. Data pada Tabel ini kemudian akan menjadi *data training* untuk keperluan perhitungan pada algoritma. *Data training*

ini bertujuan untuk meningkatkan akurasi dari algoritma, karena semakin banyak *data training* maka semakin akurat juga hasil akurasi dari algoritma.

Table	Action	Rows
<input type="checkbox"/> data	★ Browse Structure Search Insert Empty Drop	0
<input type="checkbox"/> data_container	★ Browse Structure Search Insert Empty Drop	245
<input type="checkbox"/> data_temp	★ Browse Structure Search Insert Empty Drop	1
3 tables	Sum	246

Gambar 5.5 Keseluruhan Tabel

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	id	int(5)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	id_data	int(11)			No	None		
<input type="checkbox"/> 3	us	int(5)			No	None		
<input type="checkbox"/> 4	ax	varchar(55)	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 5	ay	varchar(5)	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 6	az	varchar(5)	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 7	time	varchar(50)	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 8	in/out	enum('in', 'out')	utf8mb4_general_ci		No	None		

Gambar 5.6 Tabel Data Container

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	id_data	int(11)			No	None		
<input type="checkbox"/> 2	us	double			No	None		
<input type="checkbox"/> 3	ax	double			No	None		
<input type="checkbox"/> 4	ay	double			No	None		
<input type="checkbox"/> 5	az	double			No	None		
<input type="checkbox"/> 6	time	int(11)			No	None		
<input type="checkbox"/> 7	in/out	enum('in', 'out')	utf8mb4_general_ci		No	None		

Gambar 5.7 Tabel Data Temp

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	id_data	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	us	double			No	None		
<input type="checkbox"/> 3	ax	double			No	None		
<input type="checkbox"/> 4	ay	double			No	None		
<input type="checkbox"/> 5	az	double			No	None		
<input type="checkbox"/> 6	time	int(11)			No	None		
<input type="checkbox"/> 7	class	enum('mobil', 'motor', 'orang')	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 8	in/out	enum('masuk', 'keluar')	utf8mb4_general_ci		No	None		

Gambar 5.8 Tabel Data

## 5.2 **Conduct Daily Standup**

Kesulitan yang terjadi pada saat menjalankan aplikasi yaitu pada saat aplikasi mengklasifikasikan data, maka data tersebut akan tersimpan ke dalam basis data yang kemudian akan menjadi bagian dari data *training*. Seiring berjalannya waktu, data *training* akan semakin banyak sehingga mempengaruhi kecepatan aplikasi dalam mengolah data.

## 5.3 **Groom Prioritized Backlog**

Pada bagian ini akan membahas tentang simulasi aplikasi dan implementasi kode program yang dibuat.

### 5.3.1 **Simulasi Aplikasi**

Pada bagian ini akan dilakukan simulasi terhadap aplikasi yang dibangun sebagai berikut.

1. Data didapatkan dari sensor, yaitu sensor *accelerometer* dan sensor ultrasonik.
2. Data tersebut akan disimpan ke dalam basis data sesuai dengan urutan waktu.
3. Kemudian akan dilakukan *data-grouping*, yaitu aplikasi akan mengelompokkan data-data menjadi objek sesuai dengan kode data.
4. Kemudian akan dilakukan perhitungan menggunakan algoritma C4.5.
5. Perhitungan algoritma dimulai dari mencari *root* pohon keputusan dengan menggunakan rumus mencari entropy dengan menggunakan formula (2.2).
6. Kemudian untuk memilih atribut atau variabel sebagai *root* dapat menggunakan formula (2.1).
7. Setelah selesai mencari *root*, kemudian akan dilakukan perhitungan pada setiap cabang hingga seluruh cabang memiliki kelas yang sama.
8. Kemudian jika hasil perhitungan sudah diperoleh maka klasifikasi dapat dilakukan dengan memilih nilai akurasi terbesar.
9. Kemudian data hasil klasifikasi akan disimpan ke dalam basis data, yang kemudian akan menjadi bagian dari data *train*.
10. Data kemudian akan ditampilkan pada *interface* aplikasi.

## 5.4 Implementasi Pemrograman

Pada bagian ini akan dilakukan implementasi kode program untuk membangun aplikasi. Berikut merupakan beberapa *script* yang digunakan dalam membangun aplikasi.

### 1. Sensor Ultrasonik dan *accelerometer*

```
#define triggerPin D8
#define echoPin D7
long duration, jarak;
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  jarak = (duration/2) / 29.1;
const uint8_t scl = D6;
const uint8_t sda = D5;
double Ax,
  Ay,
  Az,
  T,
  Gx,
  Gy,
  Gz;
  Read_RawValue(MPU6050SlaveAddress,
MPU6050_REGISTER_ACCEL_XOUT_H);
  Ax = (double)AccelX/AccelScaleFactor;
  Ay = (double)AccelY/AccelScaleFactor;
  Az = (double)AccelZ/AccelScaleFactor;
  T = (double)Temperature/340+36.53;
```

### 2. REST API

```
var db = require('./DB')
const express = require('express')
const e = require('express')
const app = express() const
port = 2000
app.get('/', (req, res) => {
  res.send('<h1> Welcome to APITA </h1>')
})
app.get('/sendt/', (req, res) =>
{
  var d = new Date();
  var tyear = d.getFullYear();
  var tmon = d.getMonth();
  var tday = d.getDate();
  var thours = d.getHours();
```

```

var tmin = d.getMinutes();
var tsec = d.getSeconds();

var ax = req.query.ax
var ay = req.query.ay
var az = req.query.az
var us = req.query.us

var fulltime = tyear + "-" + (tmon+1)
+ "-" + tday + " " + thours + ":" +
tmin + ":" + tsec
console.log
("ax : " + ax + ", ay : " + ay + ", az : " + az + " us
: " + us)
var sql1 = 'select id_data, time from
data_container order by id desc limit 1'
db.query(sql1, (err, result) =>
{
    var dupcek = result[0].time.split(":")
    var dateCheck = dupcek[0].split(" ")
    var dateNow = tyear + "-" + (tmon+1) + "-" +
tday + " " + thours + ':' + tmin
    var sql2 = `INSERT INTO data_container
(id_data, us, ax, ay, az, time) VALUES
("${result[0].id_data+1}", "${us}", "${ax}", "${ay}",
"${az}", "${fulltime})`
    var sql3 = `INSERT INTO data_container
(id_data, us, ax, ay, az, time) VALUES
("${result[0].id_data}", "${us}", "${ax}", "${ay}",
"${az}", "${fulltime})`

    if(dupcek[0] + ':' + dupcek[1] == dateNow)
    {
        if(tsec > (dupcek[2] + 1)){
            db.query(sql2, (err, result2) =>
            {
                console.log('new package added')
            }
        )
        }else{
            db.query(sql3, (err, result3) =>
            {
                console.log('data added')
            }
        )
        }
    }else{
        db.query(sql2, (err, result2) =>
        {
            console.log('new package added')
        }
    )
}

```

```

    }
    console.log(result[0].id_data)
    // console.log(dupcek[2])
    res.send(dateNow)
  }
)

```

### 3. Menghitung Nilai Entropy

```

public function _hitung(array $data, array
$attributes, $base=null, $kamus=null) {
    // HITUNG JUMLAH DATA
    $jumlah_data=count($data);
    // MENGAMBIL DATA KOLOM TARGET
    $kolom_target=[];
    foreach($data as $item) {
        $kolom_target[]=$item[count($item)-1];
    }
    // MENGHITUNG NILAI ENTROPY
    $entropy_total=0;
    foreach(array_count_values($kolom_target)
as $t) {
        $entropy_total=$entropy_total -
        $t/$jumlah_data * log($t/$jumlah_data, 2);}
}

```

### 4. Menghitung entropy tiap kasus

```

foreach($attributes as $indexAttribute=> $label)
{
    $data_kolom_atribut=[]; // VARIABEL
    UNTUK MENAMPUNG DATA KOLOM ATRIBUT KE-I
    $data_kolom_target=[];
    // VARIABEL UNTUK MENAMPUNG DATA KOLOM
    TARGET
    $data_atribut_and_target=[];
    foreach ($data as $key=> $value) {

        $data_kolom_atribut[$key]=$value[$indexAttribute
-1];

        $data_kolom_target[$key]=$value[count($value)-
1];

        $data_atribut_and_target[]=$value[$indexAttribu
te-1],
            $value[count($value)-1]];
    }

    $jumlah_data_tiap_kasus=array_count_values($data_kolom
_atribut);
}

```



```

        $label_target=array_unique($this-
>getTarget());
        $total_data=0;
        $data_per_kasus=[];
        foreach($data_atribut_and_target as
$item) {
            if(
!isset($data_per_kasus[$item[0]][$item[1]])
)
                $data_per_kasus[$item[0]][$item[1]]=1;
            else
                $data_per_kasus[$item[0]][$item[1]]++;
            }
            $lx=0;
            $labels[$indexAttribute]=[];
            foreach($data_per_kasus as
$case=>$value) {
                $entropyAttribute=0;
                $l=0;
                $jumlah_case=array_sum($value);
                foreach($value as $i=>$v) {
                    $pi=$v/$jumlah_case;

                    $entropyAttribute=$entropyAttribute -
                    $pi*log($pi, 2);
                }
                if($entropyAttribute==0) {

                    $nilai_default=array_keys($value)[0];

                    $labels[$indexAttribute][$case]=$nilai_default;
                }

                $leafs[$indexAttribute][$case]=$entropyAttribute
                ;

                $lx+=$jumlah_case/$jumlah_data*$entropyAttribute
                ;
            }
            $gain=$entropy_total - $lx;
            $gains[$indexAttribute]=$gain;
        }

```

## 5. Mengurutkan *gain*

```

$l=arsort($gains)
$root=array_keys($gains)[0];
$this->rules[$root]=[];
if($base !=null) {
    $this->rules[$base][$kasus]=[ 'kasus'=>$kasus,
    'forward'=>$root];
}
foreach($leafs[$root] as $label=> $entropy) {
    if($entropy==0) {

```

```

        $this->rules[$root][$label]=[ "kasus"=>$label,
"nilai"=>$labels[$root][$label]];
        ".$labels[$root][$label], "</h1>";
    }
    if($entropy > 0 && $entropy <=1) {
    if ($base !=null) {
    $this->rules[$root][$label]=[ "kasus"=>$label.'-',
"forward"=>$base+1];
        }else {
        $this->rules[$root][$label]=$label;
        }
        ($attributes[$root]);
    $data_next_itarasi=[];
    foreach($data as $index=>$itemData) {
    if($itemData[$root - 1]==$label) {
    $data_next_itarasi[]=$itemData;
        }
    }
    $this->_hitung($data_next_itarasi,
    $attributes, $root, $label);
    }
    }
}

```

## 6. Menghitung Jumlah Kendaraan

```

<body>
<?php
include "db.php";
$sqls = mysqli_query($conn, "select * from data
where `in/out`='masuk' EXCEPT select * from
data where class = 'orang'");
    $jmlmobiltotal =
mysqli_num_rows($sqls);
$sqls2 = mysqli_query($conn, "select * from data
where `in/out`='keluar' EXCEPT select * from
data where class = 'orang'");
    $jmlmotortotal =
mysqli_num_rows($sqls2);

$sql = mysqli_query($conn, "select * from data
where class = 'mobil' AND `in/out`='masuk'");
    $jmlmobil = mysqli_num_rows($sql);
    $sql2 = mysqli_query($conn, "select * from data
where class = 'motor' AND `in/out`='masuk'");
    $jmlmotor = mysqli_num_rows($sql2);
    $sql3 = mysqli_query($conn, "select * from data
where class = 'mobil' AND `in/out`='keluar'");
    $jmlmobilout =
mysqli_num_rows($sql3);
    $sql4 = mysqli_query($conn, "select * from data
where class = 'motor' AND `in/out`='keluar'");

```

```

        $jmlmotorout =
mysql_i_num_rows($sql4);
?>
    <div class="jumbotron jumbotron-fluid">
        <div class="container-fluid br">
            <div class="container">
                <canvas id="myChart"></canvas>
            </div>
        </div>
    </div>
    <script>
        let myChart =
document.getElementById('myChart').getContext('2
d');

        // Global Options
        Chart.defaults.global.defaultFontFamily
= 'Yu Gothic';
        Chart.defaults.global.defaultFontSize =
18;
        Chart.defaults.global.defaultFontColor =
'#000';

        let massPopChart = new Chart(myChart, {
            type: 'bar', // bar, horizontalBar,
            pie, line, doughnut, radar, polarArea
            data: {
                labels: ['Cars (IN)',
'Motorcycle (IN)', 'Cars (OUT)',
'Motorcycle (OUT)'],
                datasets: [{
                    label: 'Population',
                    data: [
                        94,
                        115,
                        88,
                        106
                    ],
                    //backgroundColor:'green',
                    backgroundColor: [
                        'rgba(255, 99, 132,
0.6)',
                        'rgba(54, 162, 235,
0.6)',
                        'rgba(255, 206, 86,
0.6)',
                        'rgba(153, 102, 255,
0.6)'
                    ],
                    borderWidth: 1,
                    borderColor: '#777',

```

```

        hoverBorderWidth: 3,
        hoverBorderColor: '#000'
    }
    ],
    },
    options: {
        title: {
            display: true,
            text: 'Kendaraan masuk
keluar di De la salle',
            fontSize: 25
        },
        legend: {
            display: false,
            position: 'right',
            labels: {
                fontColor: '#000'
            }
        },
        layout: {
            padding: {
                left: 50,
                right: 0,
                bottom: 0,
                top: 0
            }
        },
        tooltips: {
            enabled: true
        }
    }
    });
</script>
<div class="container"><br><br><br>
<div class="line" style="border-left: 2px
solid #ccc;
height: 270px; position: absolute; left:
680px;"></div>

<div class="row text-center"
style="font-size: 30pt;">
<div class="col font-weight-
bold">IN</div>
<div class="col font-weight-
bold">OUT</div>
</div>
<div class="" style="position: absolute;
left: 550px; top: 648px;">
<div class="col"><h1>Today's
Report</h1></div>

</div><br>
<div class="row text-center"
style="font-size: 15pt;">

```

```

        <div class="col"><?= $jmlmobiltotal
?></div>
        <div class="col"><?= $jmlmotortotal
?></div><br><br>
        <div class="container">
            <div class="row font-weight-
bold">
                <div class="col"></div>
                <div class="col-
sm">Cars</div>
                    <div class="col-
sm">Motorcycle</div>
                        <div class="col"></div>
                            <div class="col"></div>
                                <div class="col-
sm">Cars</div>
                                    <div class="col-
sm">Motorcycle</div>
                                        <div class="col"></div>
                                            </div><br>
                                                <div class="row">
                                                    <div class="col"></div>
                                                        <div class="col-sm"><?=
$jmlmobil ?></div>
                                                            <div class="col-sm"><?=
$jmlmotor ?></div>
                                                                <div class="col"></div>
                                                                    <div class="col"></div>
                                                                        <div class="col-sm"><?=
$jmlmobilout ?></div>
                                                                            <div class="col-sm"><?=
$jmlmotorout ?></div>
                                                                                <div class="col"></div>
                                                                                    </div>
                                                                                        </div>
</div>
</body>

```

## 7. Chart Jumlah Kendaraan

```

        <canvas id="myChart" width="400"
height="400"></canvas>
        <script>
        var ctx = document.getElementById('myChart');
        var ctx =
document.getElementById('myChart').getContext('2
d');
        var ctx = $('#myChart');
        var ctx = 'myChart';
        var ctx = document.getElementById('myChart');
        var myChart = new Chart(ctx, {

```

```

        type: 'bar',
        data: {
            labels: ['Red', 'Blue', 'Yellow',
'Green', 'Purple', 'Orange'],
            datasets: [{
                label: '# of Votes',
                data: [12, 19, 3, 5, 2, 3],
                backgroundColor: [
                    'rgba(255, 99, 132, 0.2)',
                    'rgba(54, 162, 235, 0.2)',
                    'rgba(255, 206, 86, 0.2)',
                    'rgba(75, 192, 192, 0.2)',
                    'rgba(153, 102, 255, 0.2)',
                    'rgba(255, 159, 64, 0.2)'
                ],
                borderColor: [
                    'rgba(255, 99, 132, 1)',
                    'rgba(54, 162, 235, 1)',
                    'rgba(255, 206, 86, 1)',
                    'rgba(75, 192, 192, 1)',
                    'rgba(153, 102, 255, 1)',
                    'rgba(255, 159, 64, 1)'
                ],
                borderWidth: 1
            }]
        },
        options: {
            scales: {
                yAxes: [{
                    ticks: {
                        beginAtZero: true
                    }
                }]
            }
        }
    });
</script>
</body>
</html> -->
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible"
content="ie=edge">
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/Char
t.js/2.6.0/Chart.min.js"></script>

```

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstr
ap/4.5.0/css/bootstrap.min.css"
  integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYx
Ffc+NcPb1dKGj7Sk" crossorigin="anonymous">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/
3.3.7/css/bootstrap.min.css">
<title>Detail Information</title>
<style>
  body {
    background-color: #393d3a;
    color: #ccc;
  }
</style>
</head>

</html>
```

## **BAB VI**

### **PENGUJIAN**

Pada bab ini akan dilakukan pengujian terhadap aplikasi penghitung kendaraan berbasis IoT. Tahap pengujian ini merupakan bagian dari tahap ke empat dari metodologi *scrum* yaitu *review and retrospect* yang terdiri dari *convence scrum of scrum, demonstrate and validate sprint* dan *retrospect sprint*.

#### **6.1 Convence Scrum of Scrum**

Pada bagian ini akan membahas target pengujian yang akan dilakukan terhadap aplikasi yang dibangun.

##### **6.1.1 Tujuan Pengujian**

Tujuan pengujian ini dilakukan yakni untuk mengetahui apakah aplikasi yang dibangun sudah sesuai dengan perancangan yang dibuat sebelumnya, juga untuk memeriksa *error* atau *bug* pada aplikasi.

##### **6.1.2 Kriteria Pengujian**

Berikut ini merupakan kriteria pengujian yang akan menjadi tolak ukur keberhasilan aplikasi pengitung kendaraan berbasis IoT.

1. Sensor-sensor yang terpasang pada alat dapat mengirim data ke *server* ketika terdapat objek yang melintasi.
2. *Server* dapat menerima data yang dikirim dari *client*.
3. *Server* dapat melakukan klasifikasi data yang dikirim menggunakan algoritma C4.5 dan melakukan pengujian tingkat akurasi.
4. *Server* dapat menghitung jumlah kendaraan.
5. *Server* dapat memberikan informasi jumlah kendaraan yang dapat ditampilkan pada web *browser*.

##### **6.1.3 Kasus Pengujian**

Berikut ini merupakan kasus pengujian yang akan dilakukan pada aplikasi yang telah berhasil dibangun, dapat dilihat pada Tabel 6.1 halaman berikutnya.



**Tabel 6.1 Kasus Pengujian**

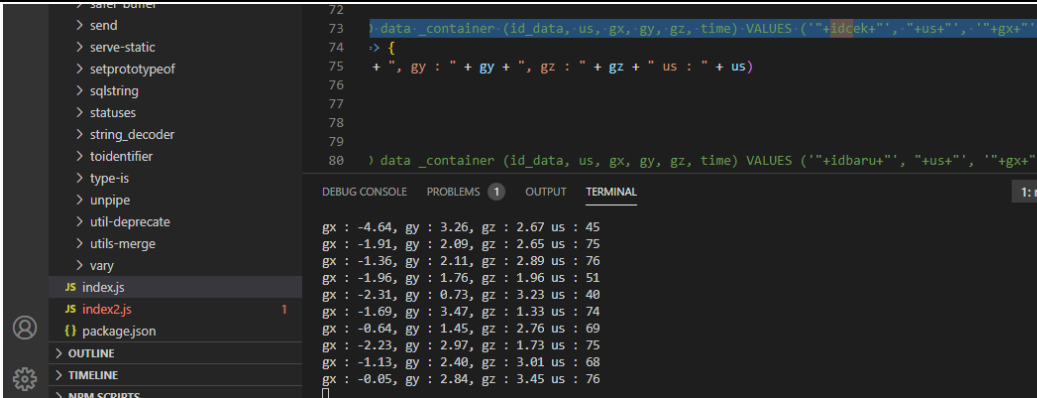
No	Pemeriksaan
1.	Apakah sensor <i>accelerometer</i> dan ultrasonik berfungsi dengan baik dan dapat mengirim data ke <i>server</i> ?
2.	Apakah ketinggian pemasangan <i>hardware</i> mempengaruhi pengambilan data?
3.	Apakah <i>server</i> dapat menerima data yang dikirimkan?
4.	Apakah fungsi <i>data-grouping</i> pada <i>server</i> berfungsi dengan baik?
5.	Apakah <i>server</i> dapat mengklasifikasi objek dengan menggunakan algoritma C4.5?
6.	Apakah <i>server</i> dapat melakukan perhitungan jumlah kendaraan berdasarkan hasil klasifikasi?
7.	Apakah <i>server</i> dapat memberikan informasi bahwa lahan parkir sudah penuh atau tidak?
8.	Apakah <i>server</i> dapat memberikan detail informasi dengan baik?
9.	Apakah <i>server</i> dapat menampilkan histori jumlah kendaraan setiap bulan dalam bentuk grafik?
10.	Apakah <i>server</i> dapat dijalankan pada <i>web browser</i> yang berbeda?

## 6.2 *Demonstrate and Validate Sprint*

Berikut ini merupakan tahapan pelaksanaan pengujian untuk mengetahui apakah aplikasi yang dibangun berjalan sesuai dengan hasil analisis dan perancangan yang sudah dilakukan pada tahap sebelumnya.


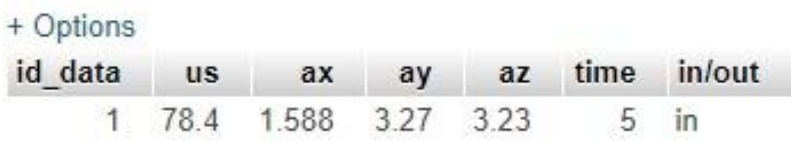

**Tabel 6.2 Tabel Pengujian**

Kegiatan Pengujian	Hasil Aktual
Melakukan pengujian apakah sensor berfungsi dengan baik dan dapat mengirim data ke aplikasi dan melakukan pengujian apakah <i>server</i> dapat menerima data yang dikirimkan.	Semua sensor bekerja dengan baik dan dapat mengirim data dengan baik serta data dapat diterima oleh <i>server</i> , yang ditunjukkan pada Gambar 6.1.

Kegiatan Pengujian	Hasil Aktual
 <pre> 72 73   data_container (id_data, us, gx, gy, gz, time) VALUES ('"+idrek"', '"+us"', '"+gx+ 74   -&gt; { 75     + ", gy : " + gy + ", gz : " + gz + " us : " + us ) 76 77 78 79 80   ) data_container (id_data, us, gx, gy, gz, time) VALUES ('"+idbaru"', '"+us"', '"+gx+ </pre> <pre> DEBUG CONSOLE  PROBLEMS 1  OUTPUT  TERMINAL gx : -4.64, gy : 3.26, gz : 2.67 us : 45 gx : -1.91, gy : 2.09, gz : 2.65 us : 75 gx : -1.36, gy : 2.11, gz : 2.89 us : 76 gx : -1.96, gy : 1.76, gz : 1.96 us : 51 gx : -2.31, gy : 0.73, gz : 3.23 us : 40 gx : -1.69, gy : 3.47, gz : 1.33 us : 74 gx : -0.64, gy : 1.45, gz : 2.76 us : 69 gx : -2.23, gy : 2.97, gz : 1.73 us : 75 gx : -1.13, gy : 2.40, gz : 3.01 us : 68 gx : -0.05, gy : 2.84, gz : 3.45 us : 76 </pre>	

**Gambar 6.1 Data Sensor**

<p>Melakukan pengujian terhadap ketinggian dari pemasangan alat.</p>	<p>Ketinggian = 0,25 meter, sensor tidak dapat menangkap seluruh permukaan objek yang lewat. Ketinggian = 0,5 meter, sensor dapat berfungsi dengan baik. Ketinggian = 0,75 meter, sensor tidak dapat menangkap seluruh permukaan objek yang lewat.</p>
<p>Melakukan pengujian terhadap jarak terbaik yang dapat ditangkap oleh sensor ultrasonik.</p>	<p>Untuk jarak yang dapat ditangkap oleh sensor sensor ultrasonik, adalah 2 cm – 4 meter. Untuk jarak terbaiknya, yaitu 5 cm sampai 2 meter dari sensor. Jika lebih dari itu, pendeteksian objek akan sedikit terganggu dikarenakan jarak yang lumayan jauh untuk dijangkau gelombang utrasonik yang dihasilkan sensor.</p>
<p>Melakukan pengujian terhadap kecepatan yang baik pada saat melewati portal.</p>	<p>Kecepatan yang disarankan untuk mobil dan motor, yaitu kurang lebih 10 kilometer per jam. Jika terlalu cepat atau terlalu lambat, maka akan mengganggu perhitungan pada</p>

Kegiatan Pengujian	Hasil Aktual
	aplikasi yang menyebabkan kesalahan klasifikasi.
Melakukan pengujian apakah fungsi <i>data-grouping</i> berjalan dengan baik.	Fitur <i>data-grouping</i> berfungsi dengan baik.
	
	
<b>Gambar 6.2 Data Grouping</b>	
Melakukan pengujian apakah <i>Server</i> dapat melakukan perhitungan dan menampilkan informasi detail.	<i>Server</i> dapat menampilkan detail informasi.
	
<b>Gambar 6.3 Informasi Detail</b>	
Melakukan pengujian apakah <i>server</i> dapat menampilkan status parkir.	Aplikasi akan menampilkan status parkir, untuk hijau berarti <i>free</i> dan merah berarti <i>full</i> . Jika jumlah kendaraan kurang dari 300 maka

Kegiatan Pengujian	Hasil Aktual
	<i>output</i> yang ditampilkan adalah <i>free</i> jika lebih atau sama dengan 300 maka <i>output</i> yang ditampilkan adalah <i>full</i> . 300 adalah kapasitas tempat parkir di dalam area kampus De La Salle manado.

**08:10:42 AM**

Parking Status

Free

**08:11:38 AM**

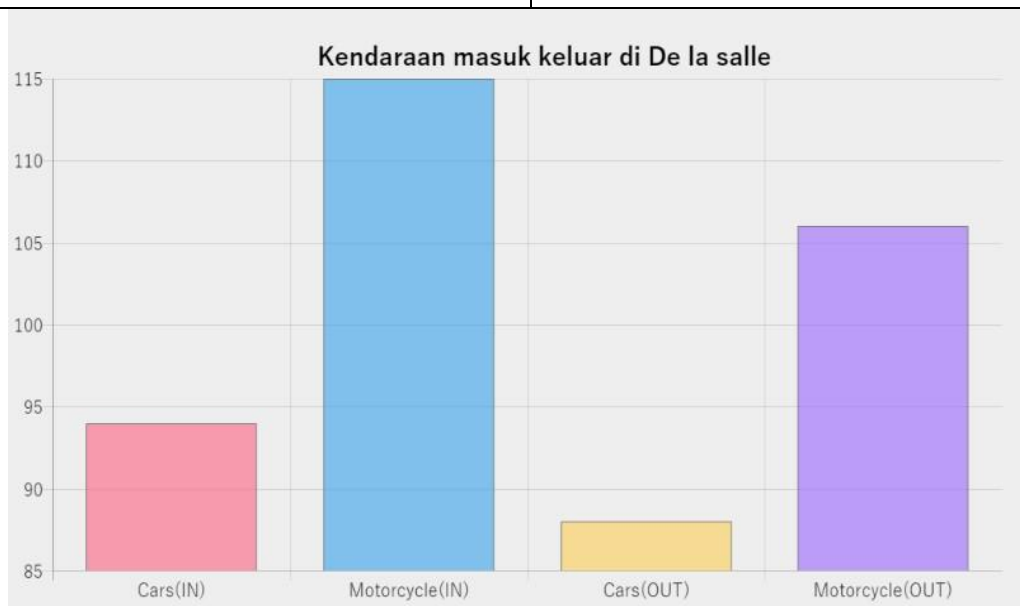
Parking Status

Full

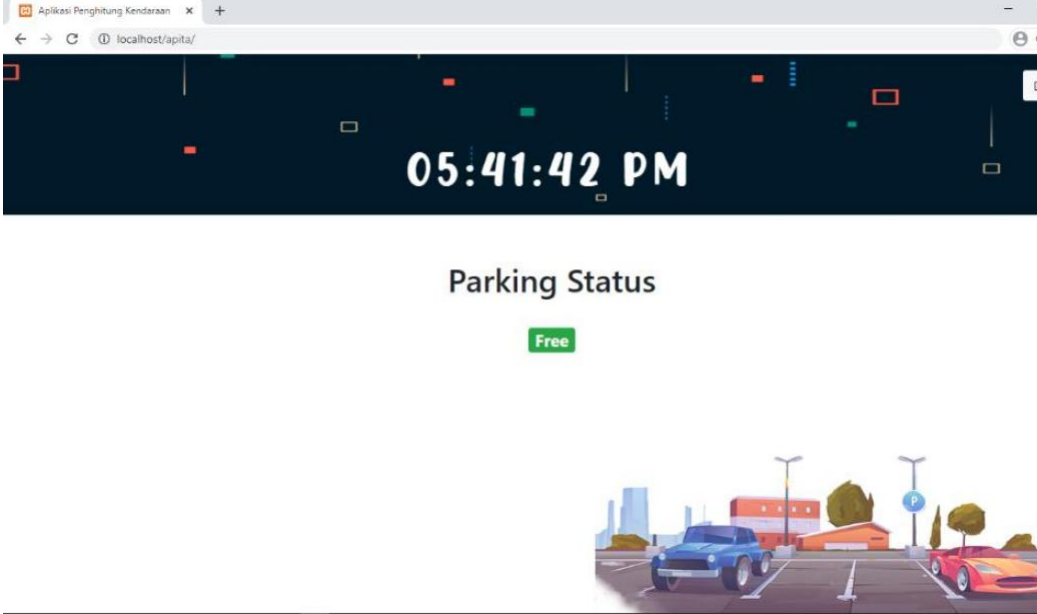
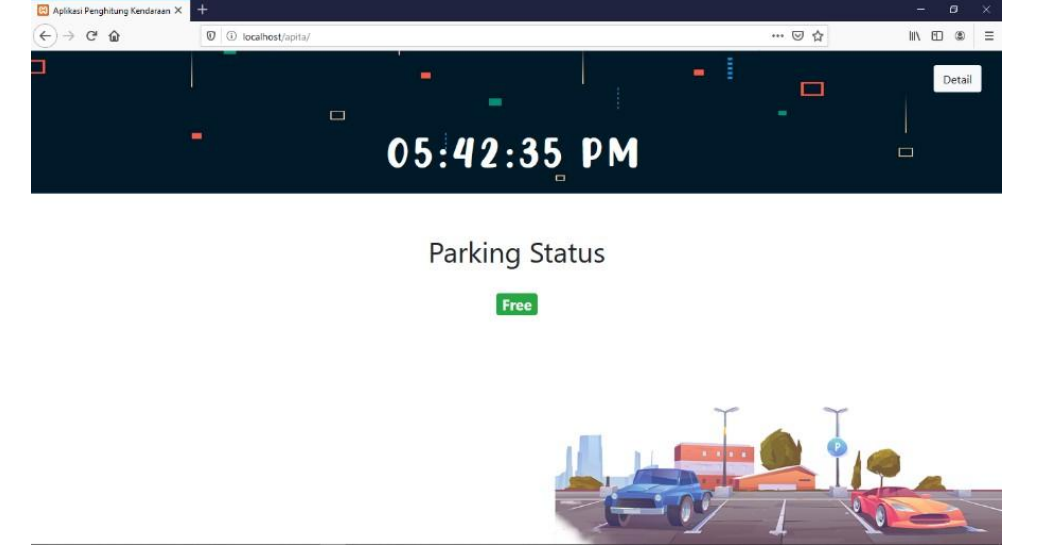
**Gambar 6.4 Status Parkir**

Melakukan pengujian apakah diagram grafik dapat berjalan dengan baik.

Diagram berjalan dengan baik.



**Gambar 6.5 Grafik**

Kegiatan Pengujian	Hasil Aktual
Menjalankan aplikasi pada <i>browser Google Chrome</i> Version 84.0.4147.89.	Aplikasi dapat berjalan tanpa masalah.
	
<b>Gambar 6.6 Pengujian Google Chrome</b>	
Menjalankan aplikasi pada <i>browser Firefox</i> version 76.0.1.	Aplikasi dapat berjalan tanpa masalah.
	
<b>Gambar 6.7 Pengujian Firefox</b>	
Menjalankan aplikasi pada <i>browser bawaan OS (Operating System)</i>	Aplikasi dapat berjalan tanpa masalah.

Kegiatan Pengujian	Hasil Aktual
Windows yaitu <i>Microsoft Edges</i> version 84.0.522.40	
	
<b>Gambar 6.8 Pengujian <i>Microsoft Edges</i></b>	

### 6.3 *Retrospect Sprint*

Semua sensor yang terpasang pada alat dapat berfungsi dengan baik yaitu mengambil dan mengirim data ke *server*, sedangkan untuk ketinggian dari pemasangan alat yang baik 0,5 meter dari permukaan tanah sedangkan ketinggian yang terburuk untuk pemasangan alat yaitu 1 meter dari permukaan tanah. Untuk jarak terbaik yang dapat dideteksi oleh sensor ultrasonik, yaitu 5 cm sampai 2 meter. Kecepatan yang disarankan saat melewati portal, yaitu kurang lebih 10 kilometer per jam. Kemudian untuk pengiriman data, *server* dapat menerima data dengan baik. Untuk fungsi *data-grouping*, dapat berfungsi dengan baik yaitu menyatukan data yang lebih dari satu menjadi satu objek utuh, sedangkan untuk klasifikasi objek, *server* dapat mengklasifikasikan objek dengan baik. *Server* juga dapat menghitung jumlah kendaraan dari hasil klasifikasi dengan baik. *Server* juga dapat memberikan informasi lahan parkir, yaitu *full* atau *free*, sedangkan untuk detail informasi, *server* dapat memprosesnya dan menampilkan ke dalam *interface* dengan baik. *Server* juga dapat memberikan informasi histori jumlah kendaraan bulanan dalam bentuk grafik dengan baik. *Server* juga dapat dijalankan di beberapa

*web browser* seperti *Google Chrome*, *Firefox* dan *Microsoft Edge* tanpa mengalami masalah yang signifikan. Kemudian hasil akurasi dari pengklasifikasian yaitu 70% yang diperoleh dari 10 kali pengujian dengan hasil 7 benar dan 3 salah yang dapat dilihat pada Tabel 6.3 di bawah.

**Tabel 6.3 Pengujian Akurasi**

Pengujian	Hasil Klasifikasi	Objek	Hasil
1	motor	mobil	salah
2	motor	motor	benar
3	orang	motor	salah
4	mobil	mobil	benar
5	mobil	mobil	benar
6	orang	orang	benar
7	motor	motor	benar
8	motor	orang	salah
9	motor	motor	benar
10	motor	motor	benar

Aplikasi telah diujicobakan kepada lima pengguna. Berdasarkan *User Acceptance Test* yang diperoleh, dapat disimpulkan bahwa aplikasi ini dapat membantu pengguna untuk mengetahui status lahan parkir, dapat mengklasifikasikan objek dengan baik, menampilkan informasi kendaraan yang keluar/masuk dan aplikasi dapat bekerja dengan baik.

## **BAB VII KESIMPULAN DAN SARAN**

### **7.1 Kesimpulan**

Berdasarkan hasil analisis dan perancangan yang telah dilakukan, dapat disimpulkan bahwa:

1. Aplikasi penghitung kendaraan berbasis IoT berhasil dibangun.
2. Aplikasi telah berhasil menerapkan algoritma C4.5 untuk mengklasifikasikan objek (mobil, motor dan orang).
3. Hasil akurasi dari pengklasifikasian objek sebesar 70%.
4. Aplikasi dapat menghitung jumlah kendaraan yang masuk dan keluar dari lahan parkir UKDLSM.
5. Aplikasi dapat membantu pengguna untuk mendapatkan informasi tentang status lahan parkir.

### **7.2 Saran**

Adapun saran untuk pengembangan aplikasi ini ke depannya, antara lain:

1. Aplikasi ini dapat diterapkan pada banyak tempat, contohnya di pusat perbelanjaan, perkantoran atau sekolah-sekolah.
2. Sensor pada aplikasi ini dapat dipadukan dengan sensor lainnya, seperti menggunakan kamera untuk pengolahan citra.
3. Menggunakan algoritma lain untuk dijadikan perbandingan atau untuk meningkatkan nilai akurasi.



## DAFTAR PUSTAKA

- [1] Admin, "unikadelasalle," universitas katolik delasalle, [Online]. Available: <https://unikadelasalle.ac.id/sejarah-ukdlsm/>. [Accessed 26 01 2020].
- [2] G. H. Cahyono, "Internet of Things," Forum Teknologi, vol. 6, no. 3, p. 37, 2015.
- [3] S. Mu'arifah, "Perancangan Sistem Penghitung Jumlah Kendaraan Pada Area Parkir Dengan Metode *Background Subtraction* Berbasis *Internet Of Things*," *TRANSIENT*, vol. 7, no. 3, p. 1, 2018.
- [4] Admin, "SCRUMstudy," VMedu.com, 2017. [Online]. Available: <https://www.scrumstudy.com/whyscrum/scrums-phases-and-processes>. [Accessed 1 2 2020].
- [5] A. R. Muzakhir; et al. "Sistem Deteksi Mobil pada Aplikasi Pembayaran Gerbang Tol Berbasis *Internet of Things*," *SMARTICS Journal*, vol. 5, no. 2, p. 97, 2019.
- [6] A. D. Limantara; et al. "Pemodelan Sistem Pelacakan Lot Parkir Kosong Berbasis Sensor *Ultrasonik* dan *Internet of things*(IoT) Pada lahan Parkir Di Luar Jalan," *jurnal.umj.ac.id*, p. 2, 2017.
- [7] Admin, "idcloudhost.com," 18 September 2019. [Online]. Available: <https://idcloudhost.com/mengenal-apa-itu-internet-of-things-iot-defenisi-manfaat-tujuan-dan-cara-kerja/>. [Accessed 23 February 2020].
- [8] Arafat, "Sistem Pengamanan Pintu Rumah Berbasis Internet," *Transient*, vol. 7, no. 3, p. 1, 2018.
- [9] Bahrin, "Sistem Kontrol Penerangan Menggunakan Arduino Uno Pada Universitas Ichsan Gorontalo," *ILKOM*, vol. 9, no. 3, p. 284, 2017.
- [10] P. S. Frima and S. A. Ridwan, "Implementasi Sensor Ultrasonik HC-SR04 Sebagai Sensor Parkir Mobil Berbasis Arduino," *Jurnal Einstein*, p. 20, 2017.
- [11] V. R. Alma, "Aplikasi Sensor Accelerometer Pada Deteksi Posisi," p. 3, 2011.
- [12] R. Abdulloh, 7 in 1 pemrograman web untuk pemula, Jakarta: PT Elex Media Komputering, 2018.
- [13] Anhar, Panduan Menguasai PHP & MySQL secara Otodidak, Jakarta: Mediakita, 2010.
- [14] V. Siahaan dan R. H. Sianipar, JavaScript Dari A Sampai Z, Medan: Sparta Publishing, 2018.
- [15] Wahana Komputer, ShortCourse Pengembangan Aplikasi *Database* berbasis JavaDB dengan Netbeans, Semarang: Penerbit Andi, 2010.
- [16] A. W. Widodo dan D. Kurnianingtyas, Sistem Basis Data, Malang: UB Press, 2017.
- [17] E. Mardiani, et al. Kumpulan Latihan SQL, Jakarta: PT Elex Media Komputering, 2016.

- [18] D. P. A. R. Hakim, et al. "Sistem Monitoring Penggunaan Air PDAM pada Rumah Tangga Menggunakan Mikrokontroler NodeMCU Berbasis Smartphone ANDROID," *Jurnal Iptek*, vol. 22, no. 2, p. 10, 2018.
- [19] Maniah dan D. Hamidin, *Analisis dan perancangan Sistem Informasi*, Yogyakarta: DEEPUBLISH, 2017.
- [20] J. Sutherland, *Panduan Scrum*: Bentang Pustaka, 2013.
- [21] N. Marpaung, "Perancangan Prototype Jemuran Pintar Berbasis Arduino Uno R3," *Riau Journal Of Computer Science*, vol. 3, no. 2, p. 73, 2017.
- [22] F. F. Harryanto dan S. Hansun, "Penerapan Algoritma C4.5 untuk Memprediksi Penerimaan Calon Pegawai Baru di PT WISE," *JATISI*, vol. 3, no. 2, p. 98, 2017.

**LAMPIRAN A**  
**USER ACCEPTANCE TESTING**

Pertanyaan	Hasil	
	Ya	Tidak
Apakah aplikasi dapat membantu pengguna untuk mengetahui status lahan parkir?		
Apakah aplikasi dapat memberikan status lahan parkir?		
Apakah aplikasi dapat mengklasifikasikan objek dengan baik?		
Apakah aplikasi dapat menampilkan detail informasi kendaraan yang keluar/masuk?		
Apakah seluruh alat dapat bekerja dengan baik?		
Apakah aplikasi dapat menerima data dari sensor dengan baik?		
Apakah aplikasi bekerja dengan baik?		

Saran:

.....  
.....  
.....

Manado, 27 Juli 2020

(Evander Kristalino)

## LAMPIRAN A

### USER ACCEPTANCE TESTING

Pertanyaan	Hasil	
	Ya	Tidak
Apakah aplikasi dapat membantu pengguna untuk mengetahui status lahan parkir?		
Apakah aplikasi dapat memberikan status lahan parkir?		
Apakah aplikasi dapat mengklasifikasikan objek dengan baik?		
Apakah aplikasi dapat menampilkan detail informasi kendaraan yang keluar/masuk?		
Apakah seluruh alat dapat bekerja dengan baik?		
Apakah aplikasi dapat menerima data dari sensor dengan baik?		
Apakah aplikasi bekerja dengan baik?		

Saran:

.....

.....

.....

Manado, 27 Juli 2020

( Daniel Kuongian)

## LAMPIRAN A

### USER ACCEPTANCE TESTING

Pertanyaan	Hasil	
	Ya	Tidak
Apakah aplikasi dapat membantu pengguna untuk mengetahui status lahan parkir?		
Apakah aplikasi dapat memberikan status lahan parkir?		
Apakah aplikasi dapat mengklasifikasikan objek dengan baik?		
Apakah aplikasi dapat menampilkan detail informasi kendaraan yang keluar/masuk?		
Apakah seluruh alat dapat bekerja dengan baik?		
Apakah aplikasi dapat menerima data dari sensor dengan baik?		
Apakah aplikasi bekerja dengan baik?		

Saran:

.....  
 .....  
 .....

Manado, 27 Juli 2020

(Hermanto Naence)

## LAMPIRAN A

### USER ACCEPTANCE TESTING

Pertanyaan	Hasil	
	Ya	Tidak
Apakah aplikasi dapat membantu pengguna untuk mengetahui status lahan parkir?		
Apakah aplikasi dapat memberikan status lahan parkir?		
Apakah aplikasi dapat mengklasifikasikan objek dengan baik?		
Apakah aplikasi dapat menampilkan detail informasi kendaraan yang keluar/masuk?		
Apakah seluruh alat dapat bekerja dengan baik?		
Apakah aplikasi dapat menerima data dari sensor dengan baik?		
Apakah aplikasi bekerja dengan baik?		

Saran:

.....  
 .....  
 .....

Manado, 27 Juli 2020

(Sweetly Mandagi)

**LAMPIRAN A**  
**USER ACCEPTANCE TESTING**

Pertanyaan	Hasil	
	Ya	Tidak
Apakah aplikasi dapat membantu pengguna untuk mengetahui status lahan parkir?		
Apakah aplikasi dapat memberikan status lahan parkir?		
Apakah aplikasi dapat mengklasifikasikan objek dengan baik?		
Apakah aplikasi dapat menampilkan detail informasi kendaraan yang keluar/masuk?		
Apakah seluruh alat dapat bekerja dengan baik?		
Apakah aplikasi dapat menerima data dari sensor dengan baik?		
Apakah aplikasi bekerja dengan baik?		

Saran:

.....  
.....  
.....

Manado, 27 Juli 2020

(Ronald Wuwungan)

## LAMPIRAN B

### KODE PROGRAM

Kode Program Arduino

```
#include <Wire.h>
#include <ESP8266WiFi.h>
#include "ESP8266HTTPClient.h"

String ssid = "bruh";
String password = "qweqweqwe";

#define triggerPin D8
#define echoPin D7
const uint8_t MPU6050SlaveAddress = 0x68;

const uint8_t scl = D6; const
uint8_t sda = D5; setting
provided in datasheet

void setup() {
  Serial.begin(9600);
  WiFi.begin(ssid, password);
  Serial.print("Connecting..");
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.print("Connected");
  Wire.begin(sda, scl);
  MPU6050_Init();
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {

long duration, jarak;
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  jarak = (duration/2) / 29.1;

  double Ax, Ay, Az, T, Gx, Gy, Gz;

  Read_RawValue(MPU6050SlaveAddress,
MPU6050_REGISTER_ACCEL_XOUT_H);
```



```

//divide each with their sensitivity scale factor
Ax = (double)AccelX/AccelScaleFactor;
Ay = (double)AccelY/AccelScaleFactor;
Az = (double)AccelZ/AccelScaleFactor;
T = (double)Temperature/340+36.53; //temperature formula
Gx = (double)GyroX/GyroScaleFactor;
Gy = (double)GyroY/GyroScaleFactor;
Gz = (double)GyroZ/GyroScaleFactor;

Serial.print("Ax: "); Serial.print(Ax);
Serial.print(" Ay: "); Serial.print(Ay);
Serial.print(" Az: "); Serial.print(Az);
Serial.print(" T: "); Serial.print(T);
  Serial.print(" Gx: "); Serial.print(Gx);
  Serial.print(" Gy: "); Serial.print(Gy);
  Serial.print(" Gz: "); Serial.println(Gz);
  Serial.print(" Jarak : "); Serial.println(jarak);
if(WiFi.status() == WL_CONNECTED){
  HTTPClient http;

  http.end();

  if(jarak <= 100){
    http.begin("http://192.168.43.227:2000/sendt/?gx=" +
String(Gx) + "&gy=" + String(Gy) + "&gz=" + String(Gz) +
"&us=" + String(jarak) );
    int httpCode = http.POST("");
    String payload = http.getString();
    Serial.println(httpCode);
    Serial.println(payload);

  }
}
delay(250);
}

```

#### Kode Algoritma

```

<?php class C45 {
  protected $data;
  protected $attributes;
  protected $target;
  protected $rules;
  protected $finalRules;
  protected $hasilPrediksi;
  public function setData(array $data) {
    $this->data=$data;
    return $this;
  }

  public function setAttributes(array $attributes) {
    $this->attributes=$attributes;
    return $this;
  }
}

```

```

protected function getTarget() {
    $target=[];
    foreach($this->data as $item) {
        $target[]=$item[count($item) -1];
    }

    return $target;
}

public function hitung() {
    $this->_hitung($this->data, $this->attributes);
    $this->generateRules();
}

public function _hitung(array $data, array $attributes,
$base=null, $kasus=null) {
    $jumlah_data=count($data);
    $kolom_target=[];
    foreach($data as $item) {
        $kolom_target[]=$item[count($item)-1];
    }
    $entropy_total=0;
    foreach(array_count_values($kolom_target) as $t) {
        $entropy_total=$entropy_total - $t/$jumlah_data *
log($t/$jumlah_data, 2);
    }
    foreach($attributes as $indexAttribute=> $label) {
        $data_kolom_atribut=[]; // VARIABEL UNTUK
MENAMPUNG DATA KOLOM ATRIBUT KE-I
        $data_kolom_target=[];
        $data_atribut_and_target=[];

        foreach ($data as $key=> $value) {
            $data_kolom_atribut[$key]=$value[$indexAttribute-1];
            $data_kolom_target[$key]=$value[count($value)-
1];
            $data_atribut_and_target[]=$value[$indexAttribute-1],
            $value[count($value)-1]];
        }

        $jumlah_data_tiap_kasus=array_count_values($data_kolom_atr
ibut);
        $label_target=array_unique($this->getTarget());
        $total_data=0;
        $data_per_kasus=[];

        foreach($data_atribut_and_target as $item) {
            if(
!isset($data_per_kasus[$item[0]][$item[1]]))
            $data_per_kasus[$item[0]][$item[1]]=1;
            else $data_per_kasus[$item[0]][$item[1]]++;
        }
    }
}

```

```

    }

    $lx=0;
    $labels[$indexAttribute]=[];

    foreach($data_per_kasus as $case=>$value) {
        $entropyAttribute=0;
        $l=0;
        $jumlah_case=array_sum($value);

        foreach($value as $i=>$v) {
            $pi=$v/$jumlah_case;
            $entropyAttribute=$entropyAttribute -
$pi*log($pi, 2);
        }

        if($entropyAttribute==0) {
            $nilai_default=array_keys($value)[0];
$labels[$indexAttribute][$case]=$nilai_default;
        }

$leafs[$indexAttribute][$case]=$entropyAttribute;

$lx+=$jumlah_case/$jumlah_data*$entropyAttribute;
    }

    $gain=$entropy_total - $lx;
    $gains[$indexAttribute]=$gain;
}

$l=arsort($gains);
$root=array_keys($gains)[0];
$this->rules[$root]=[];

if($base !=null) {
    $this->rules[$base][$kasus]=[ 'kasus'=>$kasus,
    'forward'=>$root];
}

foreach($leafs[$root] as $label=> $entropy) {
    if($entropy==0) {
        $this->rules[$root][$label]=[ "kasus"=>$label,
        "nilai"=>$labels[$root][$label]];
    }

    if($entropy > 0 && $entropy <=1) {
        if ($base !=null) {
            $this->rules[$root][$label]=[
"kasus"=>$label.'-',
            "forward"=>$base+1];
        }
    }
}

```



```

public function generateRules() {
    $l=$this->_generateRules(array_values($this->
rules) [0]);
    $x=array_keys($this->rules) [0];
    $rules[$this->attributes[$x]]=$l;
    $this->finalRules=$rules;
    // print_r($rules);
}

public $l=[];

public function _generateRules($i) {
    $l=$i;

    foreach($i as $kasmus=>$item) {
        if(isset($item['forward'])) {
            $l[$kasmus]['attribute']=$this->
attributes[$item['forward']];
            $l[$kasmus]['forward']=$this->
_generateRules($this->rules[$item['forward']]);
        }
    }

    return $l;
}

public function printRules() {
    $this->_printRules($this->finalRules);
}

public function predictDataTesting(array $data=[]) {
    $_data=array_combine($this->attributes, $data);
    $this->_predictDataTesting($_data, $this->
finalRules);
    return $this->hasilPrediksi;
}

public function _predictDataTesting(array $data, array
$rules) {
    $attribute=array_keys($rules) [0];
    $kasmusAttribute=array_values($rules) [0];

    foreach($kasmusAttribute as $kasmus=> $item) {
        if($data[$attribute]==$kasmus &&
isset($item['nilai'])) {
            $this->
hasilPrediksi=strtoupper($item['nilai']);
        }

        if($data[$attribute]==$kasmus &&
isset($item['forward']))
        {
            $l1[$item['attribute']]=$item['forward'];

```



```

var tmon = d.getMonth();
var tday = d.getDate();
var thours = d.getHours();
var tmin = d.getMinutes();
var tsec = d.getSeconds();

var ax = req.query.ax
var ay = req.query.ay
var az = req.query.az
var us = req.query.us

var fulltime = tyear + "-" + (tmon+1) + "-" + tday + "
" + thours + ":" + tmin + ":" + tsec
console.log("ax : " + ax + ", ay : " + ay + ", az : "
+ az + " us : " + us)
var sql1 = 'select id_data, time from data_container
order by id desc limit 1'
db.query(sql1, (err, result) => {
    var dupcek = result[0].time.split(":")
    var dateCheck = dupcek[0].split(" ")
    var dateNow = tyear + "-" + (tmon+1) + "-" + tday
+ " " + thours + ':' + tmin
    var sql2 = `INSERT INTO data_container (id_data,
us, ax, ay, az, time) VALUES ("${result[0].id_data+1}",
"${us}", "${ax}", "${ay}", "${az}", "${fulltime}")`
    var sql3 = `INSERT INTO data_container (id_data,
us, ax, ay, az, time) VALUES ("${result[0].id_data}",
"${us}", "${ax}", "${ay}", "${az}", "${fulltime}")`

    if(dupcek[0] + ':' + dupcek[1] == dateNow){
        if(tsec > (dupcek[2] + 1)){
            db.query(sql2, (err, result2) => {
                console.log('new package added')
            }
        )
        }else{
            db.query(sql3, (err, result3) => {
                console.log('data added')
            }
        )
        }
    }else{
        db.query(sql2, (err, result2) => {
            console.log('new package added')
        }
    )
    }
    console.log(result[0].id_data)
    // console.log(dupcek[2])
    res.send(dateNow)
}

```

```

)
Kode Program Halaman Utama
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1, shrink-to-fit=no">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/c
ss/bootstrap.min.css"
  integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYyxFfc+NcPb1d
KGj7Sk" crossorigin="anonymous">
  <title>Aplikasi Penghitung Kendaraan</title>
  <style>
    body {
      background: url('image/background2.png');
      background-repeat: no-repeat;
    }

    .clock {
      position: absolute;
      top: 20%;
      left: 50%;
      transform: translateX(-50%) translateY(-50%);
      color: #fff;
      font-size: 60px;
      font-family: CHICKEN Pie;
      letter-spacing: 7px;
    }

    button {
      position: absolute;
      right: 30px;
      top: 20px;
      z-index: 999;
    }

  </style>
</head>
<body>
<?PHP
include 'db.php';
?>

  <!-- clock -->

  <div class="jumbotron jumbotron-fluid">
    <div class="container-fluid br">
      <img src="image/jumbotronx.jpg" alt=""
width="100%" height="200"

```



```

        style="position: absolute; top: 0px;
left:0px;">
        <div id="MyClockDisplay" class="clock"
onload="showTime()"></div>
        <a href="detail.php"> <button type="button"
class="btn btn-light">Detail</button></a>
        </div>
</div>
<div class="container">
    <br><br><br><br>
    <div class="row text-center">

        <div class="col">
            <h1>
                Parking Status</h1>
        </div>
    </div>
    <br>
    <div class="row text-center">
        <div class="col">
            <?php
                $sql = mysqli_query($conn, "select * from
data");
                $jml = mysqli_num_rows($sql);

                if($jml<300){
                    echo "<h3><span class='badge badge-
success'>Free</span></h3>";
                }else{
                    echo "<h3><span class='badge badge-
danger'>Full</span></h3>";
                }
            ?>

        </div>
    </div>
</div>
<script>
    function showTime() {
        var date = new Date();
        var h = date.getHours(); // 0 - 23
        var m = date.getMinutes(); // 0 - 59
        var s = date.getSeconds(); // 0 - 59
        var session = "AM";

        if (h == 0) {
            h = 12;
        }

        if (h > 12) {
            h = h - 12;
            session = "PM";
        }
    }

```

```

        h = (h < 10) ? "0" + h : h;
        m = (m < 10) ? "0" + m : m;
        s = (s < 10) ? "0" + s : s;
        var time = h + ":" + m + ":" + s + " " +
session;

document.getElementById("MyClockDisplay").innerText =
time;

document.getElementById("MyClockDisplay").textContent =
time;

        setTimeout(showTime, 1000);

    }

    showTime();
</script>
<script src="https://code.jquery.com/jquery-
3.5.1.slim.min.js"
    integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrC
XaRkfj" crossorigin="anonymous">
</script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/
popper.min.js"
    integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvox
MfooAo" crossorigin="anonymous">
</script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/
bootstrap.min.js"
    integrity="sha384-
OgVRvuATPlz7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/
kR0JKI" crossorigin="anonymous">
</script>
</body>

</html>
        Kode Program Halaman Detail
<!-- <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

```



```

    }}
  }
}
});
</script>
</body>
</html> -->
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.6.0
/Chart.min.js"></script>
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/c
ss/bootstrap.min.css"
  integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1d
KGj7Sk" crossorigin="anonymous">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/
bootstrap.min.css">
  <title>Detail Information</title>
  <style>
    body {
      background-color: #393d3a;
      color: #ccc;
    }
  </style>
</head>

<body>
<?php
include "db.php";
$sqls = mysqli_query($conn, "select * from data where
`in/out` ='masuk' EXCEPT select * from data where class =
'orang'");
    $jmlmobiltotal = mysqli_num_rows($sqls);
    $sqls2 = mysqli_query($conn, "select * from data where
`in/out` ='keluar' EXCEPT select * from data where class =
'orang'");
    $jmlmotortotal = mysqli_num_rows($sqls2);

    $sql = mysqli_query($conn, "select * from data where class
='mobil' AND `in/out` ='masuk'");
    $jmlmobil = mysqli_num_rows($sql);
    $sql2 = mysqli_query($conn, "select * from data where
class ='motor' AND `in/out` ='masuk'");

```

```

    $jmlmotor = mysqli_num_rows($sql2);
    $sql3 = mysqli_query($conn, "select * from data where
class ='mobil' AND `in/out` ='keluar'");
    $jmlmobilout = mysqli_num_rows($sql3);
    $sql4 = mysqli_query($conn, "select * from data where
class ='motor' AND `in/out` ='keluar'");
    $jmlmotorout = mysqli_num_rows($sql4);
?>
    <div class="jumbotron jumbotron-fluid">
        <div class="container-fluid br">
            <div class="container">
                <canvas id="myChart"></canvas>
            </div>
        </div>
    </div>
    <script>
        let myChart =
document.getElementById('myChart').getContext('2d');

        // Global Options
Chart.defaults.global.defaultFontFamily = 'Yu
Gothic';
Chart.defaults.global.defaultFontSize = 18;
Chart.defaults.global.defaultFontColor = '#000';

        let massPopChart = new Chart(myChart, {
            type: 'bar', // bar, horizontalBar, pie, line,
doughnut, radar, polarArea
            data: {
                labels: ['Cars (IN)', 'Motorcycle (IN)',
'Cars (OUT)', 'Motorcycle (OUT)'],
                datasets: [{
                    label: 'Population',
                    data: [
                        94,
                        115,
                        88,
                        106
                    ],
                    //backgroundColor:'green',
                    backgroundColor: [
                        'rgba(255, 99, 132, 0.6)',
                        'rgba(54, 162, 235, 0.6)',
                        'rgba(255, 206, 86, 0.6)',
                        'rgba(153, 102, 255, 0.6)'
                    ],
                    ],
                    borderWidth: 1,
                    borderColor: '#777',
                    hoverBorderWidth: 3,
                    hoverBorderColor: '#000'
                }
            }
        }
    </script>

```

```

    },
    options: {
        title: {
            display: true,
            text: 'Kendaraan masuk keluar di De la
salle',
            fontSize: 25
        },
        legend: {
            display: false,
            position: 'right',
            labels: {
                fontColor: '#000'
            }
        },
        layout: {
            padding: {
                left: 50,
                right: 0,
                bottom: 0,
                top: 0
            }
        },
        tooltips: {
            enabled: true
        }
    }
});
</script>
<div class="container"><br><br><br>
<div class="line" style="border-left: 2px solid #ccc;
height: 270px; position: absolute; left: 680px;"></div>
<div class="row text-center" style="font-size:
30pt;">
    <div class="col font-weight-bold">IN</div>
    <div class="col font-weight-bold">OUT</div>
</div>
<div class="" style="position: absolute; left:
550px; top:648px;">
    <div class="col"><h1>Today's Report</h1></div>
</div><br>
<div class="row text-center" style="font-size:
15pt;">
    <div class="col"><?=$jmlmobiltotal ?></div>
    <div class="col"><?=$jmlmotortotal
?></div><br><br>
<div class="container">
    <div class="row font-weight-bold">
    <div class="col"></div>
    <div class="col-sm">Cars</div>

```

```

        <div class="col-sm">Motorcycle</div>
        <div class="col"></div>
        <div class="col"></div>
        <div class="col-sm">Cars</div>
        <div class="col-sm">Motorcycle</div>
        <div class="col"></div>
    </div><br>
    <div class="row">
        <div class="col"></div>
        <div class="col-sm"><?= $jmlmobil
?></div>
        <div class="col-sm"><?= $jmlmotor
?></div>
        <div class="col"></div>
        <div class="col"></div>
        <div class="col-sm"><?= $jmlmobilout
?></div>
        <div class="col-sm"><?= $jmlmotorout
?></div>
        <div class="col"></div>
    </div>
</div>
</div>
</body>
</html>

```