

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Kecelakaan**

##### **2.1.1 Kecelakaan Lalu Lintas**

Menurut Kamus Besar Bahasa Indonesia (KBBI), Kecelakaan adalah suatu kejadian atau peristiwa yang menyebabkan orang celaka.

Kecelakaan adalah suatu peristiwa di jalan yang tidak disangka-sangka dan tidak disengaja melibatkan kendaraan yang sedang bergerak dengan atau tanpa pengguna jalan lainnya, mengakibatkan korban manusia atau kerugian harta benda [4].

Kecelakaan lalu lintas merupakan suatu peristiwa pada lalu lintas di jalan yang tidak diduga dan tidak diharapkan yang sulit diprediksi kapan dan dimana terjadinya, yang melibatkan paling sedikit satu kendaraan bermotor yang dapat menyebabkan korban luka ringan atau berat, korban meninggal dunia, dan kerusakan material [4].

##### **2.1.2 Faktor Kecelakaan Lalu Lintas**

Faktor penyebab kecelakaan ada 4 (empat) faktor yaitu:

###### 1) Faktor Manusia

Faktor manusia memegang peranan yang amat dominan, karena cukup banyak faktor yang mempengaruhi perilakunya seperti pengemudi dan pejalan kaki. Karakteristik manusia sebagai host yang mempengaruhi risiko terjadinya kecelakaan lalu lintas antara lain: usia, pengalaman berkendara, perilaku berkendara, dan perilaku minum - minuman beralkohol. Selain itu juga faktor manusia yang menyebabkan terjadinya kecelakaan lalu lintas yaitu pengemudi yang mengantuk, lelah, tidak tertib dan berkecepatan tinggi [5].

###### 2) Faktor Kendaraan

Faktor kendaraan dapat menjadi faktor penyebab kecelakaan apabila kendaraan tidak dapat dikendalikan sebagaimana mestinya yaitu sebagai akibat kondisi teknis yang tidak laik. Kondisi teknis yang tidak baik jalan seperti mesin

tiba-tiba mati, rem blong, ban pecah, lampu mati dimalam hari dan lain sebagainya [5].

### 3) Faktor Jalan

Faktor jalan juga merupakan salah satu faktor penyebab terjadinya kecelakaan lalu lintas seperti kerusakan pada permukaan jalan terdapat lubang besar yang sulit dihindari pengemudi, kondisi geometrik jalan yang kurang sempurna misalnya derajat kemiringan yang terlalu kecil atau terlalu besar pada belokan [5].

### 4) Faktor Lingkungan

Kondisi lingkungan akan memberikan kontribusi terhadap kecelakaan lalu lintas. Lingkungan jalan yang kurang memadai mengakibatkan kenyamanan dari pengemudi menurun, hal ini akan berdampak kemampuan mengendalikan kendaraan akan menurun [5].

## 2.2 Kelelahan

Pengertian kelelahan secara sempit memang hanya sebatas pada lelah fisik yang dirasakan saja. Hal ini dikarenakan setiap orang yang merasakan kelelahan hanya terbatas pada keluhan-keluhan fisik yang mereka rasakan saja. Gejala yang ditimbulkan, perubahan fisik dan perasaan yang dirasakan memang berbeda pada masing-masing individu. Dari sudut pandang keselamatan kerja, medis dan *psychologi* pun memiliki definisi atau pengertian yang berbeda-beda mengenai kelelahan, tergantung dari disiplin ilmu yang dipelajari.

Menurut Saito 1999, kelelahan adalah perpaduan dari wujud penurunan fungsi mental dan fisik yang menghasilkan berkurangnya semangat kerja sehingga mengakibatkan efektifitas dan efisiensi kerja menurun [6].

Menurut Australian safety and Compensation Council 2006, Kelelahan mempengaruhi kapasitas fisik, mental, dan tingkat emosional seseorang, dimana dapat mengakibatkan kurangnya kewaspadaan, yang ditandai dengan kemunduran reaksi pada sesuatu dan berkurangnya kemampuan motorik [6].

### 2.3 Haversine Formula

*Haversine formula* digunakan untuk menghitung jarak antara dua titik di permukaan bumi yang ditentukan dengan koordinat geografis (latitude dan longitude). Formulasi ini mengambil kira-kira kurva permukaan bumi sebagai bola, bukannya datar, sehingga memberikan hasil yang lebih akurat daripada metode lain yang menganggap bumi sebagai datar [7].

Formula Haversine memerlukan nilai dua titik koordinat (latitude dan longitude), dan menggunakan jari-jari bumi ( $R$ ) sebagai parameter. Untuk menghitung jarak menggunakan formula ini, pertama-tama harus mengubah koordinat dari derajat ke radian dengan menggunakan rumus:

$$lat1 = lat1 \times (\pi/180)$$

$$lon1 = lon1 \times (\pi/180)$$

$$lat2 = lat2 \times (\pi/180)$$

$$lon2 = lon2 \times (\pi/180)$$

Kemudian, menghitung jarak menggunakan rumus:

$$h = \sin^2\left(\frac{lat2 - lat1}{2}\right) + \cos(lat1) \times \cos(lat2) \times \sin^2\left(\frac{lon2 - lon1}{2}\right)$$

$$d = 2 \times R \times \text{asin}(\text{sqrt}(h))$$

Di mana  $d$  adalah jarak antara dua titik dalam satuan yang sama dengan jari-jari bumi ( $R$ ) yang digunakan. Secara umum *Haversine formula* digunakan di bidang navigasi, survei, dan geografi, untuk menentukan jarak antar dua titik di permukaan bumi.

### 2.4 Internet of Things (IoT)

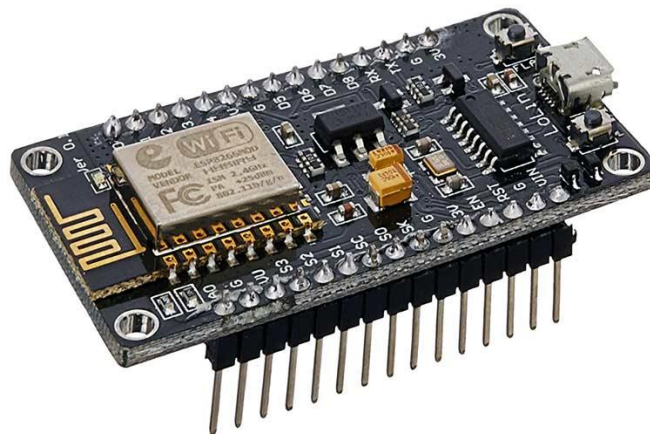
*Internet of Thing* adalah suatu revolusi teknologi yang dapat dikatakan akan mewakili masa depan komputasi dan komunikasi karena dapat menghubungkan ke perangkat tak terbatas informasi melalui internet.

*Internet of Thing* (IoT) merupakan konsep dan metode untuk kontrol jarak jauh, monitoring, pengiriman data, dan berbagai tugas lainnya. IoT akan terhubung dengan suatu jaringan sehingga dapat diakses dimana saja dan dapat mempermudah berbagai hal [8].

IoT merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terusmenerus yang memungkinkan kita untuk menghubungkan mesin, peralatan, dan benda fisik lainnya dengan sensor jaringan dan aktuator untuk memperoleh data dan mengelola kinerjanya sendiri, sehingga memungkinkan mesin untuk berkolaborasi dan bahkan bertindak berdasarkan informasi baru yang diperoleh secara independen. [9].

## 2.5 NodeMCU

NodeMCU adalah modul microcontroller berbasis ESP8266 yang menyediakan fasilitas wireless dan interkoneksi dengan internet. Ini memiliki on-board Wi-Fi, memori flash, dan modul microcontroller yang memungkinkan untuk mengeksekusi perintah dan aplikasi yang ditulis menggunakan bahasa pemrograman Lua atau C++ [10].



Gambar 2.1 NodeMCU ESP8266

NodeMCU ESP32 adalah modul microcontroller yang menggunakan chip ESP32 dari Espressif Systems. ESP32 adalah chip Wi-Fi dan Bluetooth Low Energy yang memiliki fitur yang lebih lengkap dibandingkan dengan ESP8266 yang digunakan pada NodeMCU versi sebelumnya. NodeMCU ESP32 memiliki

on-board Wi-Fi dan Bluetooth, memori flash, dan modul microcontroller yang memungkinkan untuk mengeksekusi perintah dan aplikasi yang ditulis menggunakan bahasa pemrograman Lua, C++ atau Python [11].

NodeMCU ESP32 memiliki banyak kegunaan seperti Internet of Things (IoT), Home Automation, Robotics, Prototyping, dan Penelitian dan Pengembangan, yang sama seperti NodeMCU versi sebelumnya, namun dengan tambahan fitur Bluetooth dan performa yang lebih baik.

## 2.6 Modul Stepdown

Modul step-down DC-DC adalah sebuah perangkat elektronik yang digunakan untuk mengubah tegangan dari sumber DC yang lebih tinggi menjadi tegangan yang lebih rendah. Ini dilakukan dengan menggunakan sebuah rangkaian konverter yang terdiri dari dioda, transistor, dan komponen lainnya [12].



Gambar 2.2 Modul Stepdown

Modul ini mengubah tegangan masuk yang lebih tinggi ke tegangan keluaran yang lebih rendah dengan cara mengontrol arus yang mengalir melalui komponen-komponen tersebut. Proses ini dikenal sebagai "*buck converter*" dan dapat dilakukan dengan menggunakan teknologi elektronika yang berbeda, seperti switching atau linear.

Modul step-down DC-DC sering digunakan dalam aplikasi seperti sistem pengisian baterai, alat-alat elektronik portabel, dan sistem pembangkit listrik tenaga surya. Dalam aplikasi pengisian baterai, modul ini digunakan untuk menurunkan tegangan dari sumber listrik AC atau DC yang lebih tinggi menjadi tegangan yang cocok untuk pengisian baterai. Dalam alat-alat elektronik portabel, modul ini digunakan untuk mengubah tegangan dari baterai yang lebih tinggi menjadi tegangan yang sesuai untuk alat tersebut. Dalam sistem pembangkit listrik tenaga surya, modul ini digunakan untuk mengubah tegangan dari panel surya yang lebih tinggi menjadi tegangan yang cocok untuk digunakan oleh beban.

Modul step-down DC-DC juga dilengkapi dengan berbagai fitur perlindungan seperti perlindungan arus lebih, perlindungan tegangan lebih, dan perlindungan pendek-sirkuit untuk melindungi komponen dari kerusakan.

## 2.7 Resistor

Resistor adalah komponen elektronika pasif yang membatasi arus listrik yang mengalir melalui suatu rangkaian elektronik. Resistor memiliki tahanan atau resistansi listrik yang tetap, sehingga arus listrik yang mengalir melalui resistor akan terbatas dan diubah menjadi panas [13].



Gambar 2.3 Resistor

Fungsi utama resistor adalah membatasi atau mengontrol besaran arus dan tegangan dalam suatu sistem elektronik, seperti mengontrol arus listrik agar tidak

melebihi batas yang diinginkan, membagi tegangan, memelihara tahanan, dan melindungi komponen lain dalam sistem.

Ketahanan resistor dipengaruhi oleh ukuran dan bahan dasar yang digunakan dan biasanya dalam skala ohm ( $\Omega$ ) atau megaohm ( $M\Omega$ ). Resistor memiliki peran penting dalam sistem elektronik dan membantu menjaga kestabilan dan keselamatan sistem serta memastikan bahwa komponen lain dalam sistem bekerja dengan benar.

## **2.8 Sensor GPS**

GPS adalah sistem navigasi yang menggunakan satelit yang didesain agar dapat menyediakan posisi secara instan, kecepatan dan informasi waktu di hampir semua tempat di muka bumi, setiap saat dan dalam kondisi cuaca apapun.

GPS merupakan sebuah alat atau sistem yang dapat digunakan untuk menginformasikan penggunanya dimana dia berada (secara global) di permukaan bumi yang berbasis satelit. Data dikirim dari satelit berupa sinyal radio dengan data digital [14].

GPS adalah sistem navigasi yang berbasis satelit yang saling berhubungan yang berada di orbitnya. Satelit-satelit itu milik Departemen Pertahanan (*Departemen of Defense*) Amerika Serikat yang pertama kali diperkenalkan mulai tahun 1978 dan pada tahun 1994 sudah memakai 24 satelit [15].

Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang diberi nama GPS receiver yang berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. Posisi di ubah menjadi titik yang dikenal dengan nama *Waypoint* yang nantinya akan berupa titik-titik koordinat lintang dan bujur dari posisi seseorang atau suatu lokasi kemudian di layar pada peta elektronik [14].

## **2.9 Aki Motor / Accu Motor**

Aki motor adalah sebuah baterai yang digunakan untuk menyediakan arus listrik yang diperlukan oleh sistem starter motor dan sistem kelistrikan lainnya pada

sepeda motor. Aki motor terdiri dari sel-sel baterai yang disusun bersama-sama dan diikat dengan elektrolit yang digunakan untuk menghasilkan arus listrik [16].



Gambar 2.4 Aki Motor / Accu Motor

Aki motor dapat dibagi menjadi dua jenis utama: aki bekas cuci (lead-acid) dan aki tanam (sealed maintenance-free). Aki bekas cuci memerlukan perawatan yang lebih sering dan harus diisi ulang dengan elektrolit, sedangkan aki tanam tidak memerlukan perawatan khusus dan tidak memerlukan pengisian ulang.

Kondisi aki yang baik dapat ditentukan dengan melakukan tes voltase. Tes voltase ini dilakukan dengan mengukur tegangan baterai saat ini. Jika tegangan baterai di bawah 12 volt, maka baterai harus diganti.

Faktor-faktor yang dapat menyebabkan aki motor rusak antara lain over charging, under charging, dan pemakaian yang berlebihan. Oleh karena itu, penting untuk melakukan perawatan yang tepat pada aki motor agar dapat digunakan dalam jangka waktu yang lama.

### **2.10 Flowchart**

*Flowchart* adalah penggambaran grafis dari langkah-langkah program dan urutan proses. Flowchart membantu analis dan programmer memecah masalah menjadi



potongan-potongan kecil dan membantu menganalisis alternatif lain yang sedang berjalan.

#### 1) *Flowchart* Sistem

*Flowchart* sistem adalah diagram yang menunjukkan alur kerja atau operasi yang dilakukan di seluruh sistem dan menjelaskan urutan proses dalam sistem. Dengan kata lain, *flowchart* adalah gambaran grafis dari urutan kombinasi program yang membentuk sistem [17].

*Flowchart* sistem terdiri dari data yang mengalir melalui sistem dan proses yang mengubah data tersebut. Data dan proses dalam *flowchart* sistem dapat digambarkan secara *online* (langsung terhubung ke komputer) atau *offline* (tidak langsung terhubung ke komputer, seperti mesin tik, mesin kasir, atau kalkulator).

#### 2) *Flowchart* Dokumen

*Flowchart* Dokumen melacak aliran data yang ditulis melalui sistem. Bagan alur alur kerja administrasi sering disebut sebagai bagan alur dokumen.

Tujuan utamanya adalah untuk melacak aliran formulir sistem dan laporan dari satu bagian ke bagian lain dan bagaimana aliran formulir dan laporan diproses, dicatat, dan disimpan.

#### 3) *Flowchart* Skematik

*Flowchart* skema mirip dengan diagram alir sistem yang menggambarkan sistem atau proses. *Flowchart* skematis ini tidak hanya menggunakan simbol *flowchart* standar, tetapi juga gambar komputer, periferal, tabel, atau perangkat lain yang digunakan dalam sistem.

*Flowchart* skematik digunakan sebagai alat komunikasi antara analis sistem dan mereka yang tidak terbiasa dengan notasi bagan alur tradisional. Menggunakan gambar daripada simbol diagram alir menghemat waktu yang dihabiskan orang untuk mempelajari simbol abstrak sebelum mereka dapat memahami diagram alir.

Gambar-gambar ini mengurangi kemungkinan terjadinya salah tafsir terhadap sistem yang disebabkan oleh kurangnya pemahaman terhadap simbol-simbol yang digunakan. Gambar juga memudahkan pengamat untuk memahami apa yang dimaksud analis, sehingga hasilnya lebih menyenangkan dan bebas dari kesalahpahaman.

#### 4) *Flowchart* Program

*Flowchart* program dihasilkan dari bagan alur sistem. *Flowchart* program menggambarkan secara lebih rinci bagaimana setiap langkah dari suatu program atau proses sebenarnya dilakukan. *Flowchart* ini menunjukkan setiap langkah dari suatu program atau proses dalam urutan yang tepat di mana mereka terjadi.

Programmer menggunakan *flowchart* program untuk menggambarkan urutan instruksi untuk program komputer. Analis sistem menggunakan *flowchart* program untuk menggambarkan urutan tugas kerja dalam suatu proses atau operasi.

#### 5) *Flowchart* Proses

*Flowchart* proses adalah teknik gambar teknik industri yang menguraikan dan menganalisis langkah selanjutnya dalam suatu program atau sistem.

*Flowchart* proses digunakan oleh insinyur industri untuk meneliti dan mengembangkan proses manufaktur. Dalam analisis sistem, *flowchart* ini dapat digunakan secara efektif untuk melacak alur laporan atau formulir.

### **2.11 *Object Oriented Programming* (OOP)**

*Object Oriented Programming* (OOP) adalah paradigma baru dalam rekayasa perangkat lunak berbasis objek dan kelas. Para ahli mengakui bahwa berorientasi objek saat ini merupakan metodologi terbaik dalam rekayasa perangkat lunak. Orientasi Objek melihat perangkat lunak sepotong demi sepotong, menggambarkan bagian dalam suatu objek.

Teknologi objek menganalogikan dengan sistem aplikasi berorientasi objek dalam kehidupan nyata. Dengan demikian, keunggulan teknologi objek adalah model yang dibuat akan sangat mirip dengan dunia nyata, yang permasalahannya

akan diselesaikan oleh sistem yang dibangun. Model objek, properti dan pemrosesannya dapat diambil langsung dari objek dunia nyata [18].

### **2.12 Unified Modeling Language (UML)**

*Unified Modeling Language* (UML) adalah alat untuk memvisualisasikan dan mendokumentasikan hasil analisis dan desain yang mencakup sintaks dari sistem pemodelan visual. Ini juga merupakan seperangkat konvensi pemodelan yang digunakan untuk mendefinisikan atau mendeskripsikan sistem perangkat lunak sehubungan dengan objek.

Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang sudah ada sebelumnya yaitu Grady Booch, OOD (*Object Oriented Design*), Jim Rumbaugh, OMT (*Object Modeling Technique*) dan Ivar Jacobson OOSE (*Object Oriented Software Engineering*) [18].

Pada UML edisi kedua ini terdiri dari tiga kategori, dan terdapat 13 macam diagram, yaitu:

#### a. Struktur Diagram

Menguraikan elemen spesifikasi dimulai dengan kelas, objek, dan hubungan mereka, dan kemudian dilanjutkan dengan mendokumentasikan arsitektur logis dari sistem. Struktur diagram dalam UML terdiri atas :

##### 1) *Class Diagram*

*Class diagram* menggambarkan struktur statis kelas dalam sistem, dan menggambarkan atribut, operasi, dan hubungan antar kelas. *Class diagram* membantu memvisualisasikan struktur kelas sistem dan merupakan jenis diagram yang paling banyak digunakan. Selama fase desain, *class diagram* berperan dalam menangkap struktur semua kelas yang membentuk arsitektur sistem yang dibuat.

## 2) *Object Diagram*

*Object diagram* mengilustrasikan kejelasan dan pewarisan kelas, dan terkadang digambar saat merencanakan kelas, atau untuk membantu pemangku kepentingan non-pemrograman yang mungkin menganggap diagram kelas terlalu abstrak.

## 3) *Component Diagram*

*Component diagram* menjelaskan struktur fisik kode, memetakan tampilan logis kelas proyek ke kode aktual yang mengimplementasikan logika tersebut.

## 4) *Deployment Diagram*

Deployment diagram menguraikan arsitektur fisik perangkat lunak, perangkat keras, dan artefak sistem. Deployment diagram dapat dianggap sebagai akhir dari ruang lingkup kasus penggunaan, yang menggambarkan bentuk fisik sistem daripada diagram konseptual tentang bagaimana pengguna dan perangkat berinteraksi dengan sistem.

## 5) *Composite Structure Diagram*

Composite structure diagram mirip dengan class diagram, tetapi menggambarkan bagian individu daripada seluruh kelas. Kita dapat menambahkan konektor untuk menghubungkan dua atau lebih bagian dalam asosiasi atau ketergantungan.

## 6) *Package Diagram*

Package diagram sering digunakan untuk menggambarkan organisasi tingkat tinggi dari proyek perangkat lunak. Atau dengan kata lain, buat grafik ketergantungan paket untuk setiap paket di pohon model.

### b. *Behavior Diagram*

Menggambarkan ciri-ciri *behavior*/metode/ fungsi dari sebuah sistem atau *business process*. *Behavior diagram* dalam UML terdiri atas :

7) *Use case diagram*

Diagram yang mendeskripsikan aktor, *use case*, dan hubungannya sebagai urutan operasi yang memberikan nilai terukur bagi aktor. Kasus penggunaan direpresentasikan sebagai elips horizontal dalam use case diagram penggunaan UML.

8) *Activity diagram*

Menggambarkan aktivitas, objek, status, transisi status, dan peristiwa. Dengan kata lain, diagram alur kerja aktivitas menggambarkan perilaku sistem sehubungan dengan aktivitas

9) *State machine diagram*

Menggambarkan state, transisi state dan event.

c. *Interaction Diagram*

10) *Communication Diagram*

Mirip dengan sequence diagram, tetapi communication diagram juga digunakan untuk memodelkan perilaku dinamis dari use case. Dibandingkan dengan sequence diagram, communication diagram lebih fokus menunjukkan kolaborasi objek daripada time series.

11) *Interaction Overview Diagram*

Interaction overview diagram berfokus pada penggambaran aliran kontrol interaksi, dan node adalah interaksi atau peristiwa interaksi.

12) *Sequence Diagram*

Sequence diagram menjelaskan interaksi objek secara kronologis. Singkatnya, diagram urutan adalah deskripsi langkah demi langkah termasuk urutan

kronologis (urutan) perubahan logis yang harus dialami sesuatu berdasarkan use case diagram.

### 13) *Timing Diagram*

Timing diagram di UML didasarkan pada diagram urutan perangkat keras yang awalnya dikembangkan oleh insinyur listrik.