

## **BAB II**

### **TEORI PENUNJANG**

#### **2.1 Surat Izin Mengemudi (SIM)**

SIM (Surat Izin Mengemudi) adalah kartu tanda bukti registrasi dan identifikasi yang diterbitkan oleh Polri dan diberikan kepada seseorang/pengguna kendaraan bermotor yang telah memenuhi persyaratan administrasi, kesehatan jasmani dan rohani, mampu memahami tata tertib dan peraturan lalulintas, serta terampil dalam mengemudikan kendaraan bermotor. SIM wajib dimiliki oleh semua pengemudi kendaraan bermotor (roda dua maupun lebih). Ada banyak persyaratan dan tahapan yang harus dipenuhi untuk memperoleh SIM, tentunya tidak mudah dan tidak sembarangan orang untuk memperolehnya.

Secara umum, untuk memperoleh SIM, pemohon wajib lulus dari tiga tahapan penting, antara lain tes kesehatan (terutama mata), mengisi soal/ujian teori tentang pengetahuan lalu-lintas, dan ujian praktik tentang teknis dan keterampilan penguasaan kendaraan bermotor; untuk ujian dengan kendaraan roda 2 atau lebih biasanya menggunakan simulator/kendaraan yang sebenarnya tergantung dari instansi pada daerah masing-masing. Pihak kepolisian akan menyerahkan SIM setelah pemohon memenuhi syarat administrasi dan mampu melalui tahap demi tahap dengan poin kelulusan yang baik.

SIM, selain sebagai surat izin mengemudi juga dapat digunakan sebagai pengganti sementara bukti identitas diri (KTP= Kartu Tanda Penduduk) untuk berbagai urusan perbankan dan urusan pemerintahan. Sebagai salah satu identitas penting, SIM memuat berbagai informasi penting sebagai berikut:

1. Nama pemilik.
2. Tempat dan tanggal lahir pemilik.
3. Alamat pemilik.
4. Pekerjaan pemilik.

5. Tinggi badan pemilik.
6. Tempat dan tanggal diterbitkan.
7. Nama dan cap instansi yang menerbitkan.
8. Nama dan tanda tangan pejabat yang menerbitkan.
9. Golongan dan nomor surat izin mengemudi.
10. Jenis surat izin mengemudi.
11. Tanggal berakhir masa berlaku.
12. Tanda tangan dan sidik jari pemilik.
13. Pas foto dari pemilik.[4]

## 2.2 Haversine Formula

Rumus Haversine adalah persamaan penting dalam navigasi, memberikan jarak lingkaran besar antara dua titik pada bola dari garis bujur dan garis lintangnya.[5] Penggunaan rumus ini mengasumsikan pengabaian efek ellipsoidal, cukup akurat untuk sebagian besar perhitungan, juga pengabaian ketinggian bukit dan kedalaman lembah di permukaan bumi. Berikut adalah rumus haversine.[6]

$$x = (\text{long2} - \text{long1}) * \cos ((\text{lat2} + \text{lat1}) \div 2)$$

$$y = (\text{lat2} - \text{lat1})$$

$$d = \sqrt{x * x + y * y} * R$$

Keterangan Rumus

$x$  = Longitude / Lintang

$y$  = Latitude / Bujur

$d$  = Jarak

$R$  = Radius Bumi = 6371 km

1 derajat = 0,0174532925 radian

### 2.2.1 Penggunaan Formula Haversine

Formula Haversine dapat digunakan untuk mencari perhitungan jarak antar dua titik, umumnya pengukuran jarak akan dilakukan dengan menarik garis lurus pada peta untuk diukur jarak antar keduanya.

Tabel 2.1 Contoh Kasus

Titik A (Gedung Sate)		Titik B (Masjid Lembang)	
Long: 107.618633	Lat: -6.901361	Long: 107.618279	Lat: -6.811771

Berdasarkan data dari Tabel 2.1, terdapat koordinat awal dan koordinat tujuan, yang akan dijadikan bahan ukur untuk pencarian jarak haversine

Pada Gambar 2.1 merupakan ilustrasi pencarian jarak menggunakan metode haversine. terlihat pada Google Maps jarak yang didapat jika menarik garis lurus dari Titik A menuju Titik B adalah 10,05 Km.



Gambar 2.1 Ilustrasi Pengukuran

Setelah didapat hasil pengukuran google maps pada Gambar 2.1, berikut adalah penjelasan perhitungan jarak menggunakan formula haversine yang akan dikerjakan di Microsoft Windows Excel, hasilnya sebagai berikut :

Perhitungan Jarak melalui koordinat			
	Latitude	Longitude	
Gedung Sate	-6.901361	107.618633	
Mesjid Lembang	-6.811771	107.618279	
Harvesine =	9.96	Km	=(6371.1*((2*ASIN(SQRT((SIN((RADIANS(B7)-RADIANS(B6))/2)^2)+COS(RADIANS(B7))*COS(RADIANS(B6))*(SIN((RADIANS(C7)-RADIANS(C6))/2)^2))))))

Gambar 2.2 Contoh Perhitungan Metode Haversine

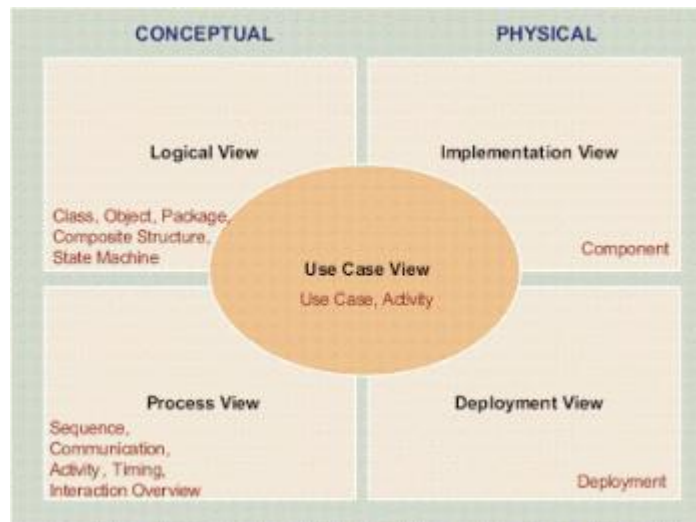
Perhitungan jarak menggunakan metode haversine pada Gambar 2.2, didapati bahwa jarak antara 2 titik adalah 9,96 Km.[7]

### 2.3 Unified Modeling Language (UML)

UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifacts (bagian dari informasi yang digunakan untuk dihasilkan oleh proses pembuatan perangkat lunak, artifact tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi object. UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera Rational Software Corps. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.[8]

### 2.3.1 Bagian – bagian UML

Bagian-bagian utama dari UML adalah view, diagram, model element, dan general mechanism.



Gambar 2.3 View UML

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. Beberapa Jenis view dalam UML antara lain : use case view, logical view, component view, concurrency view, dan deployment view.

1. Use case View Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan external actors. Actor yang berinteraksi dengan sistem dapat berupa user atau sistem lainnya.
2. View ini digambarkan dalam use case diagrams dan kadang-kadang dengan activity diagrams. View ini digunakan terutama untuk pelanggan, perancang (designer), pengembang (developer), dan pengujian sistem (tester).
3. Logical View Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (class, object, dan relationship) dan kolaborasi dinamis yang terjadi ketika object mengirim pesan ke object lain dalam suatu fungsi tertentu. View ini digambarkan dalam class diagrams untuk struktur statis dan dalam state, sequence, collaboration, dan activity diagram untuk model

dinamisnya. View ini digunakan untuk perancang (designer) dan pengembang (developer).

4. Component View Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari code module diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administrative lainnya. View ini digambarkan dalam component view dan digunakan untuk pengembang (developer).
5. Concurrency View Membagi sistem ke dalam proses dan prosesor. View ini digambarkan dalam diagram dinamis (state, sequence, collaboration, dan activity diagrams) dan diagram implementasi (component dan deployment diagrams) serta digunakan untuk pengembang (developer), pengintegrasian (integrator), dan pengujian (tester).
6. Deployment View Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (nodes) dan bagaimana hubungannya dengan lainnya. View ini digambarkan dalam deployment diagrams dan digunakan untuk pengembang (developer), pengintegrasian (integrator), dan pengujian (tester).[9]

### **2.3.2 Diagram UML**

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu view tertentu dan ketika digambarkan biasanya dialokasikan untuk view tertentu. Adapun jenis diagram antara lain :

1. Use Case Diagram Menggambarkan sejumlah external actors dan hubungannya ke use case yang diberikan oleh sistem. Use case adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari use case symbol namun dapat juga dilakukan dalam activity diagrams. Use case digambarkan hanya yang dilihat dari luar oleh actor (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam sistem.

2. Class Diagram Menggambarkan struktur statis class di dalam sistem. Class merepresentasikan sesuatu yang ditangani oleh sistem. Class dapat berhubungan dengan yang lain melalui berbagai cara: associated (terhubung satu sama lain), dependent (satu class tergantung/menggunakan class yang lain), specialized (satu class merupakan spesialisasi dari class lainnya), atau package (grup bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa class diagram.
3. State Diagram Menggambarkan semua state (kondisi) yang dimiliki oleh suatu object dari suatu class dan keadaan yang menyebabkan state berubah. Kejadian dapat berupa object lain yang mengirim pesan. State class tidak digambarkan untuk semua class, hanya yang mempunyai sejumlah state yang terdefinisi dengan baik dan kondisi class berubah oleh state yang berbeda.
4. Sequence Diagram Menggambarkan kolaborasi dinamis antara sejumlah object. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.
5. Collaboration Diagram Menggambarkan kolaborasi dinamis seperti sequence diagrams. Dalam menunjukkan pertukaran pesan, collaboration diagrams menggambarkan object dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan sequence diagrams, tapi jika penekannya pada konteks gunakan collaboration diagram.
6. Activity Diagram Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti use case atau interaksi.
7. Component Diagram Menggambarkan struktur fisik kode dari komponent. Komponent dapat berupa source code, komponent biner, atau executable component. Sebuah komponent berisi informasi tentang logic class atau

class yang diimplementasikan sehingga membuat pemetaan dari logical view ke component view.

8. Deployment Diagram Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan hubungan komputer dengan perangkat (nodes) satu sama lain dan jenis hubungannya. Di dalam nodes, executable component dan object yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh node tertentu dan ketergantungan komponen.[10]

#### **2.4 HyperText Markup Language (HTML)**

HTML (Hyper Text Markup Language) adalah sebuah bahasa formatting yang digunakan untuk membuat sebuah halaman website. Di dalam dunia pemrograman berbasis website(Web Programming), HTML menjadi pondasi dasar pada halaman website. sebuah file HTML di di simpan dengan ekstensi .html (dot html). dan dapat di eksekusi atau diakses menggunakan web browser(Google Chrome, Mozilla Firefox, Opera, Safari dan lain-lain). seperti yang sudah di jelaskan, HTML adalah dasar dari sebuah website. untuk membuat sebuah website tidak cukup hanya menggunakan HTML, kita memerlukan bantuan CSS, JavaScript dan PHP untuk membuat sebuah website yang dinamis. jika halaman website dibuat hanya menggunakan HTML saja maka halaman website tersebut di sebut halaman statis karena tidak memiliki aksi atau fungsifungsi yang dapat mengelola website. tentu developer akan sangat di sibukkan dengan harus mengubah lagi file HTML setiap ingin mengupdate artikel.[11]

#### **2.5 Hypertext Preprocessor (PHP)**

PHP Pertama kali ditemukan pada 1995 oleh seorang Software Developer bernama Rasmus Lerdorf. Ide awal PHP adalah ketika itu Radmus ingin mengetahui jumlah pengunjung yang membaca resume onlinenya. script yang dikembangkan baru dapat melakukan dua pekerjaan, yakni merekam informasi



visitor, dan menampilkan jumlah pengunjung dari suatu website. Dan sampai sekarang kedua tugas tersebut masih tetap populer digunakan oleh dunia web saat ini. Kemudian, dari situ banyak orang di milis mendiskusikan script buatan Rasmus Lerdorf, hingga akhirnya rasmus mulai membuat sebuah tool/script, bernama Personal Home Page (PHP).

Kebutuhan PHP sebagai tool yang serba guna membuat Lerdorf melanjutkan untuk mengembangkan PHP hingga menjadi suatu bahasa tersendiri yang mungkin dapat mengkonversikan data yang di inputkan melalui Form HTML menjadi suatu variable, yang dapat dimanfaatkan oleh sistem lainnya. Untuk merealisasikannya, akhirnya Lerdorf mencoba mengembangkan PHP menggunakan bahasa C ketimbang menggunakan Perl. Tahun 1997, PHP versi 2.0 di rilis, dengan nama Personal Home Page Form Interpreter (PHP-FI). PHP Semakin populer, dan semakin diminati oleh programmer web dunia.

Rasmus Lerdorf benar-benar menjadikan PHP sangat populer, dan banyak sekali Team Developer yang ikut bergabung dengan Lerdorf untuk mengembangkan PHP hingga menjadi seperti sekarang, Hingga akhirnya dirilis versi ke 3-nya, pada Juni 1998, dan tercatat lebih dari 50.000 programmer menggunakan PHP dalam membuat website dinamis.

Pengembangan demi pengembangan terus berlanjut, ratusan fungsi ditambahkan sebagai fitur dari bahasa PHP, dan di awal tahun 1999, netcraft mencatat, ditemukan 1.000.000 situs di dunia telah menggunakan PHP. Ini membuktikan bahwa PHP merupakan bahasa yang paling populer digunakan oleh dunia web development. Hal ini mengagetkan para developernya termasuk Rasmus sendiri, dan tentunya sangat diluar dugaan sang pembuatnya. Kemudian Zeev Suraski dan Andi Gutschman selaku core developer (programmer inti) mencoba untuk menulis ulang PHP Parser, dan diintegrasikan dengan menggunakan Zend scripting engine, dan mengubah jalan alur operasi PHP. Dan semua fitur baru tersebut di rilis dalam PHP 4. Fungsi bahasa pemrograman php sendiri untuk web digunakan untuk dapat menyesuaikan tampilan konten sesuai dengan situasi. Web yang bersifat

dinamis juga digunakan untuk menyimpan data ke database dengan memproses form dan juga dapat mengubah tampilan website sesuai inputan dari seorang user.

13 Juli 2004, evolusi PHP, PHP telah mengalami banyak sekali perbaikan disegala sisi, dan wajar jika netcraft mengumumkan PHP sebagai bahasa web populer didunia, karena tercatat 19 juta domain telah menggunakan PHP sebagai server side scriptingnya. PHP saat ini telah Mendukung XML dan Web Services, Mendukung SQLite. Tercatat lebih dari 19 juta domain telah menggunakan PHP sebagai server scriptingnya. Benarbenar PHP sangat mengejutkan. Yang menjadikan PHP berbeda dengan HTML adalah proses dari PHP itu sendiri. HTML merupakan bahasa statis yang apabila kita ingin merubah konten/isinya maka yang harus dilakukan pertama kali nya adalah, membuka file-nya terlebih dahulu, kemudian menambahkan isi kedalam file tersebut. Beda hal nya dengan PHP. Bagi anda yang pernah menggunakan CMS seperti wordpress atau joomla yang dibangun dengan PHP tentunya, ketika akan menambahkan konten kedalam website, anda tinggal masuk kedalam halaman admin, kemudian pilih new artikel untuk membuat halaman/content baru. Artinya hal ini, seorang user tidak berhubungan langsung dengan scriptnya. Sehingga seorang pemula sekalipun dapat menggunakan aplikasi seperti itu[12]

## **2.6 My Structured Query Language (MySQL)**

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Tidak seperti PHP atau Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB.[13]

## 2.7 Google Maps API

Google Maps Application Programming Interface atau yang biasa disebut dengan Google Maps API, Google Maps API adalah sebuah layanan (service) yang diberikan oleh Google kepada para pengguna untuk memanfaatkan Google Maps dalam mengembangkan aplikasi. Google Maps API menyediakan beberapa fitur untuk memanipulasi peta, dan menambah konten melalui berbagai jenis services yang dimiliki.[14] Google Maps API adalah library JavaScript. Tools ini disediakan secara gratis untuk penggunaan basic dimana hanya akan diberikan akses dalam penggunaan Google Maps. Terdapat banyak aplikasi yang kita gunakan sehari-hari yang menggunakan layanan Google Maps antara lainnya aplikasi ojek online seperti Go-Jek, GIS Persebaran Madrasah, dan aplikasi tracking barang online.

### 2.7.1 Parameter pada Google Maps API

Layanan Google Maps API ini menyediakan beberapa parameter tampilan yaitu :

a. Roadmap

Roadmap adalah parameter tampilan peta geografis dua dimensi, ini adalah tampilan default dari Google Maps ketika akan digunakan.

b. Satellite

Satellite merupakan citra geografis berupa tampilan satelit.

c. Terrain

Menunjukkan relief permukaan fisik bumi, pada parameter ini tampilan akan mencitrakan sebuah tekstur bumi tinggi atau rendahnya akan terlihat dari semakin pekat warna pada tampilannya.

d. Hybrid

Hybrid merupakan tampilan gabungan antara Satellite dan Roadmap, sehingga pengguna dapat melihat tampilan peta dari satelit, serta dapat melihat rute jalan yang ada.

## 2.8 Codeigniter

Codeigniter adalah sebuah web application framework yang bersifat open source digunakan untuk membangun aplikasi php dinamis. Tujuan utama pembangunan Codeigniter adalah untuk membantu developer untuk mengerjakan aplikasi lebih cepat daripada menulis semua code dari awal. Codeigniter menyediakan berbagai macam library yang dapat mempermudah dalam pembangunan. Codeigniter diperkenalkan kepada public pada tanggal 28 Februari 2006.

Codeigniter sendiri dibangun menggunakan konsep Model-View-Controller development pattern. Codeigniter sendiri merupakan salah satu framework tercepat dibandingkan dengan framework lainnya. Pada acara frOSCon (August 2008). pembuat php Rasmus Lerdorf mengatakan dia menyukai codeigniter karena dia lebih ringan dan cepat dibandingkan framework lainnya ("because it is faster, lighter and the least like a framework.") [15]

## 2.9 Skala Likert

Skala likert adalah skala pengukuran yang dikembangkan oleh Likert (1932). Skala likert mempunyai empat atau lebih butir-butir pertanyaan yang dikombinasikan sehingga membentuk sebuah skor/nilai yang merepresentasikan sifat individu, misalkan pengetahuan, sikap, dan perilaku. Dalam proses analisis data, komposit skor, biasanya jumlah atau rata-rata, dari semua butir pertanyaan dapat digunakan [16] Secara umum setiap pilihan jawaban responden akan memiliki bobot angka, berikut adalah contohnya :

1. Jawaban Sangat (SS) Setuju memiliki poin 5.
2. Jawaban Setuju (S) memiliki poin 4.
3. Jawaban Netral (N) memiliki poin 3.
4. Jawaban Tidak Setuju (TS) memiliki poin 2.
5. Jawaban Sangat Tidak Setuju (STS) memiliki poin 1.

Setelah semua responden menjawab semua pertanyaan yang diajukan maka baru terlihat dengan memasukkan persamaan dibawah ini :

$$Rumus = T * Pn$$

Keterangan

T = Total Jumlah responden yang memilih.

Pn = Pilihan angka skor likert.

Langkah selanjutnya adalah mencari rumus index, karena perlunya melihat rentan persentase dalam setiap pilihan jawabannya, misal dari 100 suara akan menjadi seperti berikut :

1. Angka rentang 0-19,9% = Sangat Tidak Setuju(STS).
2. Angka rentang 20-39,9% = Tidak Setuju(TS).
3. Angka rentang 40-59,9% = Netral (N).
4. Angka rentang 60-79,9% = Setuju (S).
5. Angka rentang 80-100% = Sangat Setuju (SS)

Langkah terakhir adalah melakukan penyelesaian akhir dengan persamaan sebagai berikut :

$$Rumus = Total Skor / Skor Maksimal * 100$$