BAB II
LANDASAN TEORI

2.1 Landasan Teori

2.1.1 Rancang Bangun

Rancang bangun merupakan proses dalam membangun sebuah sistem guna terciptanya sistem atau aplikasi baik dalam bentuk yang baru maupun mengganti sistem sebelumnya atau menyempurnakan sistem yang telah ada, tentunya dapat dilakukan secara menyeluruh atau setengah sistem[3].

Dalam proses merancang perlu adanya analisis agar dapat memecahkan suatu masalah, kemudian ini yang disebut sebagai teknik pemecahan masalah. Dalam hal ini perlu memasukkan sebuah proses penambahan, penghapusan dan perubahan pada bagian penting yang terdapat di sistem[4].

2.1.2 Kegiatan Patroli

Patroli adalah penugasan dari pimpinan untuk melaksanakan kegiatan pemantauan atau pengawasan, baik di wilayah perkotaan, pedesaan maupun jalan umum yang menjadi tujuan lokasi dari kegiatan tersebut[5]. Dengan adanya patroli dapat meminimalisir angka kejahatan, pelanggaran dan kecelakaan lalu lintas, Kepolisian sebagai aparat penegak hukum dapat mengemban tugas salah satunya patroli untuk menjelajahi daerah atau beberapa tempat yang dianggap tidak aman dan nyaman bagi masyarakat[5].

2.1.3 Metode *Prototyping*

Metode *prototyping* merupakan konsep model yang digunakan dalam pengembangan perangkat lunak. Dalam hal ini menghasilkan sebuah *prototype* sebagai demonstrasi awal dari sistem yang akan diciptakan, adapun penerapannya perlu melibatkan hubungan antara pengembang sistem dengan pengguna sebagai bentuk komunikasi agar dapat melakukan identifikasi masalah yang dihadapi, perencanaan dan analisis kebutuhan, pemodelan, pembangunan *prototype* serta penyerahan dan penelitian sistem kepada pengguna[2].

2.1.4 *Website*

Website merupakan bagian dari teknologi berupa kumpulan halaman dimuat dalam bentuk situs yang memiliki kegunaan untuk mengirim informasi dan data antar pengguna melalui bantuan internet[6]. Terdapat komponen yang ada didalam sebuah *website*, seperti teks, audio, gambar, video, animasi, data dalam bentuk tabel dan lain-lain.

2.1.5 Laravel

Laravel merupakan salah satu *framework* dari bahasa pemrograman PHP yang bisa digunakan secara *open source* dengan menerapkan konsep MVC pada susunan kode programnya[7]. Umumnya digunakan untuk pembuatan *website* secara dinamis dimana *website* yang menggunakan laravel dapat berinteraksi langsung terhadap manajemen data dan informasi.

2.1.6 *Application Programming Interface*

Application Programming Interface atau disingkat API merupakan komunikasi melalui permintaan dan pengiriman dalam data dengan sebutan *request* dan *respons*[8], standar permintaan dan pengiriman data yang digunakan oleh API adalah menggunakan JSON untuk melakukan transaksi data.

Dalam implementasinya, API tidak berinteraksi dengan pengguna melainkan berperan sebagai penghubung data antara sisi *client*, seperti aplikasi dan *website* dengan sisi sistem basis data, untuk cara kerjanya yaitu *client* memiliki *request* data dari keinginan pengguna melalui API kemudian diterima oleh sistem basis data, data yang diinginkan akan dikirim kepada *client* melalui *respons* dan pengguna bisa mengelola data tersebut dari sisi *client*[8].

2.1.7 *Single Page Application*

Single Page Application atau disingkat SPA merupakan teknik memuat halaman secara cepat tanpa waktu yang lama yaitu kurang dari 5 detik, penerapan SPA dapat mempercepat dalam menampilkan konten, informasi dan data tanpa perlu menunggu proses *request* dari *server* hingga selesai[9].

SPA memungkinkan untuk meringankan beban *server* dalam melayani *request* dan *respons* dari *client*, sehingga pengguna dapat merasakan kecepatan *loading website* ketika ingin memperoleh konten, informasi dan data.

2.1.8 *Unified Modeling Languange*

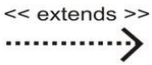
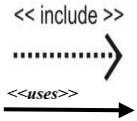
Unified Modeling Languange atau disingkat UML merupakan teknik pemodelan untuk membuat skema atau merancang alur dari pengembangan perangkat lunak[11]. Pemodelan pada UML menggunakan bahasa visual, dimana penggambaran dan dokumentasi menggunakan pendekatan yang berorientasi pada objek[12], sehingga memudahkan dalam mengembangkan sistem yang menggunakan pemrograman berorientasi objek.

A. Use Case Diagram

Use Case Diagram adalah jenis diagram UML yang digunakan untuk menggambarkan interaksi atau hubungan antara aktor dengan sistem, diagram ini memvisualisasikan skema fungsional dan tahapan proses bisnis dalam sistem, termasuk bagaimana aktor berasosiasi dengan sistem dan fungsi yang bisa dilakukan oleh setiap aktor. Berikut penjelasan simbol *Use Case Diagram* dijelaskan pada Tabel 2.1.

Tabel 2. 1 *Use Case Diagram*

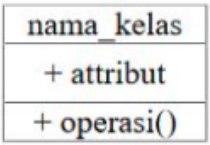
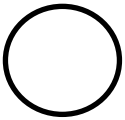
No	Simbol	Nama	Keterangan
1		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*.
2		*Actor*	Menspesifikasikan himpunan peran yang pengguna mainkan ketika ber*int*eraksi dengan *Use case*.
3		*Association*	Untuk menunjukan adanya komunikasi antara aktor dan *Use case* yang berpartisipasi pada *Use case* atau *Use case* memiliki *int*eraksi dengan aktor.

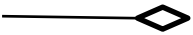
No	Simbol	Nama	Keterangan
4		*Extend*	Simbolyang digunakan untuk menunjukan adanya relasi *Use Case* tambahan ke sebuah *Use Case* dimana *Use Case* yang ditambahkan dapat berdiri sendiri walaupun tanpa *Use Case* tambahan tersebut.
5		*Generalization*	Hubungan generalisasi dan spesialisasi antara dua buah *Use Case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6		*Include / Uses*	Relasi *Use case* tambahan ke sebuah *Use case* dimana *Use case* ini untuk menjalankan fungsi atau sebagai syarat dijalankanya *Use case* ini.

B. Class Diagram

Class Diagram adalah salah satu jenis diagram dalam *Unified Modeling Language* yang digunakan untuk menggambarkan struktur statis dari sebuah sistem dan mendeksripsikan sistem kedalam jenis – jenis objek[13]. Diagram ini menampilkan kelas-kelas yang ada dalam sistem beserta hubungan diantara kelas-kelas tersebut. Setiap kelas dalam *class diagram* diwakili oleh kotak – kotak *diagram* yang berisi atribut, *method* dan relasi antar setiap kelas[14]. Berikut penjelasan untuk simbol-simbol pada *class diagram* pada Tabel 2.2.

Tabel 2. 2 *Class Diagram*

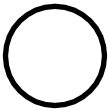
No	Simbol	Nama	Keterangan
1		*Kelas*	Kelas pada stuktur sistem
2		*Interface*	Sama dengan konsep interface dalam pemrograman berorientasi objek
3		*Association*	Untuk menunjukan adanya komunikasi atau relasi antar kelas
4		*Directed Association*	Sebagai penghubung antar kelas dengan makna kelas yang atau digunakan oleh kelas yang lain, dan juga disertai dengan multiplicity.

No	Simbol	Nama	Keterangan
5		*Generalisasi*	Hubungan antar kelas dengan makna dari umum ke khusus
6		*Dependency*	Hubungan antar kelas dengan makna kebergantungan dengan kelas lain.
7		*Agregasi*	Hubungan antar kelas dengan makna semua-bagian

2.1.9 *Flowchart*

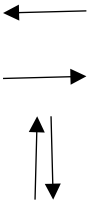
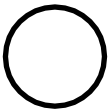
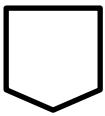
Flowchart merupakan teknik penggambaran atau langkah-langkah yang dilakukan berurut dan terdokumentasi untuk mendeskripsikan proses bisnis dalam sistem secara sederhana dan jelas[15]. *Flowchart* memiliki 3 jenis simbol, diantaranya simbol *input-output*, simbol penghubung dan simbol proses. Berikut penjelasan simbol -simbol pada *flowchart* dapat dilihat pada tabel 2.3.

Tabel 2. 3 Simbol *input-output*

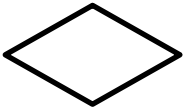
No	Simbol	Nama	Fungsi
1		*Input-output*	Simbol yang digunakan untuk menyatakan proses *input* dan *output* tanpa tergantung dengan jenis peralatanya.
2		*Punched Card*	Simbol yang digunakan untuk menyatakan *input* berasal dari kartu atau *output* ditulis ke kartu.
3		*Magnetic-Tape Unit*	Simbol yang menyatakan *input* berasal dari pita *magnetic* atau *output* disimpan ke pita *magnetic*.
4		*Disk Storage*	Simbol yang menyatakan *input* berasal dari *disk* atau *output* disimpan ke *disk*.

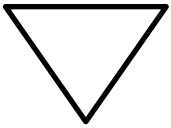
No	Simbol	Nama	Fungsi
5		*Document*	Simbol yang digunakan untuk mencetak laporan ke *printer.*
6		*Keying Operation*	Simbol yang menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai *keyboard*.
7		*Display*	Simbol yang digunakan untuk menampilkan informasi yang relevan.

Tabel 2. 4 Simbol penghubung

No	Simbol	Nama	Fungsi
1		Arus atau *flow*	Menyatakan jalan arus atau proses.
2		*Communication Link*	Menyatakan adanya transisi suatu data atau informasi dari satu lokasi ke lokasi lainnya.
3		*Connector*	Simbol yang berfungsi untuk sambungan dari proses ke proses lainnya pada halaman yang sama.
4		*Off Line Connector*	Menyatakan sambungan dari proses ke proses lainnya berbeda halaman.

Tabel 2. 5 Simbol proses

No	Simbol	Nama	Fungsi
1		Proses	Simbol yang menunjukan setiap pengolahan yang akan dilakukan oleh komputer.
2		Manual	Simbol yang digunakan untuk menyatakan suatu proses yang tidak dilakukan oleh komputer.
3		*Decision* (Logika)	Simbol yang digunakan untuk suatu kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau pilihan.

No	Simbol	Nama	Fungsi
4		*Predefine* Proses	Simbol yang digunakan untuk menyediakan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
5		*Terminal*	Simbol yang digunakan untuk menyatakan awal atau akhir suatu program.
6		*Keying Operation*	Simbol yang menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai *keyboard*.
7		*Off Line Storage*	Simbol yang menyatakan data dalamsimbol akan disimpan ke suatu media tertentu.

No	Simbol	Nama	Fungsi
8		*Manual Input*	Simbol yang digunakan untuk memasukan data secara manual dengan menggunakan *online keyboard.*

2.1.10 Desain Antarmuka

Desain antarmuka adalah bagian visual dari sistem atau perangkat fisik yang menentukan bagaimana pengguna akan berinteraksi dengan sistem atau perangkat fisik tersebut serta bagaimana data dan informasi akan ditampilkan[16]. Tujuan dari desain antarmuka adalah untuk meningkatkan kegunaan sebuah sistem dan pengalaman pengguna ketika berinteraksi dengan sistem maupun perangkat fisik.

2.1.11 *My Structured Query Languange*

My Structured Query languange atau disingkat MySQL merupakan basis penyimpanan data yang menggunakan tipe data relasional[17]. data pada MySQL tersimpan didalam tabel yang terstruktur, rapi dan sederhana sehingga mudah dan cepat dalam mengelolanya, baik untuk sistem skala kecil maupun skala besar, basis data ini bersifat *open source* dan telah memiliki komunitas yang besar.

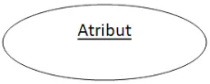
2.1.12 *Entity Relationship Diagram*

Entity Relationship Diagram atau disingkat ERD merupakan suatu metode dimana pemodelan database yang digunakan adalah model konseptual, yaitu jenis model data semantik dari sistem. Sistem yang digunakan dalam relasi entitas adalah database relasional top-down[10]. Diagram yang digunakan merupakan gambaran model hubungan entitas yang disebut diagram hubungan entitas.

Entitas ini merupakan suatu objek yang dibedakan berdasarkan hubungan yang teridentifikasi secara unik yang menghubungkan satu sama lain, sedangkan atribut-atributnya membentuk sifat-sifat setiap entitas menurut beberapa konvensi, dalam penggunaannya[10]. ERD berbentuk notasi grafis yang digunakan untuk membuat database yang menghubungkan data ke data lain. Berikut penjelasan simbol -simbol pada ERD dapat dilihat pada tabel 2.6.

Tabel 2.6 Simbol ERD

No	Simbol	Nama	Fungsi
1		Entitas (*entity*)	Merupakan data yang disimpan melalui tabel database, objek yang menyimpan data dan perlu menyimpan data tersebut agar aplikasi komputer dapat mengaksesnya. Penamaan entitas biasanya lebih mengarah pada kata benda dan belum termasuk nama tabel

No	Simbol	Nama	Fungsi
2		Atribut (*attribute*)	Merupakan properti dari relasi entitas atau yang memberikan deskripsi mendetail tentang entitas atau relasi, atribut terdiri kolom data atau kolom yang harus disimpan dalam entitas.
3		Relasi (*Relationship*)	Merupakan hubungan entitas yang menunjukkan adanya hubungan antara beberapa entitas dari kumpulan entitas yang berbeda.

No	Simbol	Nama	Fungsi
4		Garis (*Line/Connector*)	Garis sebagai penghubung antara relasi dan kumpulan entitas serta kumpulan entitas dan atributnya. Garis dapat memudahkan pengguna melihat dan memahami alur suatu ERD, sehingga bagian awal dan akhir terlihat jelas.