

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Berikut ini beberapa skripsi dan jurnal penelitian terdahulu yang sudah ada, yaitu:

Pongki Hartoto, Merry Agustina, Suryayusra, 2015. Website Informasi Arus Lalu Lintas Pada Satuan Lalu Lintas Polresta Palembang. *Website* ini merupakan suatu sistem yang bertujuan untuk mempermudah masyarakat untuk mengetahui informasi tentang kemacetan lalu lintas di kota Palembang yang dapat dioperasikan melalui web. Sistem pemantauan arus lalu lintas sebagian kecil terdiri dari beberapa perangkat keras seperti komputer, dan beberapa komponen elektronik lainnya. Sedangkan bahasa pemrograman yang digunakan adalah *PHP, HTML, MySQL, CSS* dan beberapa komponen – komponen pendukung lainnya yang digunakan berdasarkan fungsi masing – masing. Aplikasi yang dibuat hanya menampilkan titik yang dipasang cctv dan menampilkan aktifitas yang tertangkap cctv^[1].

Penelitian diatas memiliki kesamaan dengan penelitian yang dilakukan oleh peneliti yaitu menggunakan bahasa pemrograman *PHP, HTML, MySQL*. Perbedaan penelitian yaitu jurnal diatas menampilkan aktifitas-aktifitas yang tertangkap cctv sedangkan penelitian yang dilakukan peneliti menampilkan informasi waktu lampu hijau dan merah setiap simpang, kondisi antrian setiap simpang, dan informasi dari twitter yang diberikan para pengguna jalan akan tertampil di web.

Retno Bambang Sudibyo, 2013. Pembangunan Sistem Informasi Lalu Lintas Berbasis Web Direktorat Lalu Lintas Kepolisian Daerah Kalimantan Timur. Pembangunan sistem informasi ini berbasis *web* kota Balikpapan. Hal ini bisa dilihat dari penambahan jumlah penduduk yang cepat dan juga perkembangan fasilitas – fasilitas umum. Selain itu juga, kota Balikpapan yang banyak terdapat daerah industri menyebabkan banyak arus lalu lintas. Untuk itu diperlukan sistem informasi lalu lintas yang handal seperti informasi tentang kemacetan dan kecelakaan. Sistem informasi harus dapat diakses secara

akurat, mudah, luas, *user friendly*, dan juga dapat diupdate databasenya setiap saat. Penelitian ini bertujuan untuk membuat rancangan sistem informasi lalu lintas kota Kalimantan yang berbasis web. Untuk menghindari masalah yang berhubungan dengan lisensi perangkat lunak, digunakan perangkat lunak opensource yaitu PHP, Macromedia Dreamweaver dan MySQL^[2].

Penelitian diatas memiliki kesamaan dengan penelitian yang dilakukan oleh peneliti yaitu menggunakan bahasa pemograman PHP, HTML, MySQL dan databse dapat diupdate setiap saat. Perbedaan penelitian yaitu jurnal diatas menampilkan aktifitas-aktifitas yang tertangkap cctv sedangkan penelitian yang dilakukan peneliti menampilkan informasi waktu lampu hijau dan merah setiap simpang, kondisi antrian setiap simpang, dan informasi dari twitter yang diberikan para pengguna jalan akan tertampil di web.

2.2 Landasan Teori

2.2.1 Website

Website atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, atau gabungan dari semuanya baik bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait yang masing-masing dihubungkan dengan jaringan-jaringan halaman. Hubungan antara satu halaman web dengan halaman web yang lainnya disebut hyperlink, sedangkan teks yang dijadikan media penghubung disebut *hypertext*^[3].

Jenis – jenis web berdasarkan sifatnya :

1. *Website* Dinamis

Sebuah *website* yang menyediakan content atau isi yang selalu berubah – ubah setiap saat. Bahasa pemograman yang digunakan anantara lain *PHP*, *ASP*, *.NET* dan memanfaatkan database *MySQL* atau *MS SQL*.

2. *Website* Statis

Merupakan *website* yang jarang diubah. Bahasa pemograman yang digunakan adalah *HTML* dan belum memanfaatkan *database*.

2.2.2 HTML

HTML atau *HyperText Markup Language* adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web internet dan permformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi^[4].

Salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan dihalaman web Halaman ini dikenal sebagai *web page*. Dokumen HTML merupakan dokumen yang disajikan pada *web browser*. Kode HTML

```
<HTML>
```

```
</HTML>
```

Masing-masing baris di atas disebut tag. Tag adalah kode yang digunakan untuk me-mark-up (memoles) teks *ASCII* menjadi file *HTML*. Setiap teks diapit dengan tanda kurung runcing. Ada tag pembuka yaitu `<HTML>` dan adan tag penutup yaitu `</HTML>` yang ditandai dengan tanda slash (garis miring) didepan awal tulisannya. Tag diatas memberikan kaidah bahwa yang akan ditulis diantara kedua tag tersebut adalah isi dari dokumen HTML.

2.2.3 PHP

PHP singkatan dari PHP : *Hypertext Preprocessor* yaitu bahasa pemograman *web server-side* yang bersifat *open source*. PHP merupakan script yang terintergrasi dengan HTML dan berada pada server (server side HTML embedded scripting). PHP adalah script yang digunakan untuk membuat web dinamis. Dinamis berarti halaman yang ditampilkan dibuat saat halaman itu diminta oleh client. Mekankisme ini menyebabkan informasi yang diterima client selalu terbaru atau up to date. Semua script PHP dieksekusi pada server di mana script tersebut dijalankan. PHP tergolong sebagai perangkat lunak *opersource* yang diatur dalam aturan *general purpose license* (GPL)^[5].

2.2.4 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread* dan multiuser. Dan diperkirakan sekitar 6 juta instalasi diseluruh dunia. MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah license komersial untuk kasus – kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. MySQL berfungsi mengolah database menggunakan bahasa SQL. MySQL bersifat opensource sehingga bisa menggunakannya secara gratis^[5].

2.2.5 AJAX

Ajax adalah singkatan dari *Asynchronous JavaScript* dan *XML*. Ajax merupakan teknik baru didunia pengembang halaman web, yang lebih kaya dibandingkan web biasa. Ajax memungkinkan sebuah halaman web memperbaharui data dari sever yang ditampilkannya tanpa harus melakukan *refresh*, sehingga halaman terlihat lebih responsif.

Fitur dan ciri utama daru ajax adalah sebuah halaman wb yang tidak perlu di *refresh* setiap kali ada data baru yang ingin ditampilkan. Dengan demikian sebuah halaman web akan terasa seperti aplikasi desktop. Kecepatan, interaktivitas, fungsionalitas, dan tingkat kegunaan halaman web akan meningkat.

2.2.6 UML

Unified Modeling Language (UML) digunakan untuk melakukan pemodelan sistem/perangkat lunak dengan menggunakan *tools* yang ada. Dengan pemodelan menggunakan UML, rekayasa dan pengembangan perangkat dapat dilakukan dengan fokus pengembangan dan desain perangkat lunak terhadap :

1. Tinjauan umum bagaimana arsitektur sistem secara keseluruhan.
2. Penelaah bagaimana objek-objek dalam sistem saling mengirimkan pesan (message) dan saling bekerjasama satu sama lain.

3. Menguji apakah sistem/perangkat lunak sudah berfungsi seperti yang seharusnya.
4. Dokumentasi sistem/perangkat lunak untuk keperluan-keperluan tertentu di masa yang akan datang.



Setiap sistem yang kompleks seharusnya bisa dipandang dari sudut pandang yang berbeda-beda sehingga bisa dilakukan pemahaman secara menyeluruh. Dalam upaya-nya tersebut, UML menyediakan sembilan jenis diagram yang dapat dikelompokkan berdasarkan sifatnya yang statis ataupun dinamis.






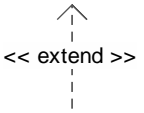
Terdapat berbagai diagram dalam merancang sistem menggunakan UML, diantaranya adalah :



1. Use-Case Diagram

Use Case Diagram secara grafis menggambarkan interaksi antara sistem, sistem eksternal, dan pengguna. Dengan kata lain *Use Case Diagram* secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna (*user*) mengharapkan interaksi dengan sistem itu. *Use Case* secara naratif digunakan secara tekstual untuk menggambarkan urutan langkah-langkah dari setiap interaksi. Dalam pembicaraan tentang *Use Case*, pengguna biasanya disebut aktor. Aktor adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem^[6].

Tabel 2.1 Simbol *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Use Case</i>	Deskripsi berdasarkan keperluan aktor, merupakan “apa” yang dikerjakan sistem, bukan “bagaimana” sistem mengerjakannya. <i>Use case</i> dibuat

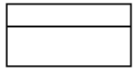

			berdasarkan keperluan actor.
3		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4		<i>Association 1 arah</i>	Mengindikasikan bila actor berinteraksi secara <i>pasif</i> dengan system anda
5		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
7		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
8		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use</i>



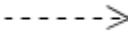


			case sumber pada suatu titik yang diberikan.
9		<i>System Boundary</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

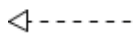
2. Class Diagram

Class diagram bersifat statis tetapi sering pula memuat kelas-kelas aktif dan memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain^[6].

Tabel 2.2 Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
2		<i>Directional Association</i>	Asosiasi ini menggambar bahwa pesan atau urutan kejadian terjadi dari hanya salah satu kelas sedangkan kelas yang


			lain pasif
3		<i>Bidirectional Association</i>	Asosiasi ini terjadi ketika salah satu kelas mengirimkan pesan kepada kelas yang lain kemudian kelas yang lain mengirimkan pesan kepada kelas yang mengirimnya pesan.
4		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
5		<i>Dependency</i>	Relasi jenis ini menunjukkan bahwa sebuah kelas mengacu kepada kelas lainnya ¹ . Oleh sebab itu perubahan pada kelas yang diacu akan sangat berpengaruh pada kelas yang mengacu
6		<i>Agregation</i>	Suatu bentuk relasi yang jauh lebih kuat dari asosiasi. Agregasi dapat diartikan bahwa suatu kelas merupakan bagian dari kelas yang lain namun bersifat tidak wajib.
7		<i>Composite</i>	Merupakan relasi yang paling kuat

			dibandingkan asosiasi dan agregasi. Komposisi diartikan bahwa suatu kelas merupakan bagian yang wajib dari kelas yang lain.
8		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek. Realisasi, bisa disebut juga implementasi merupakan suatu relasi yang menunjukkan penerapan terhadap suatu interface kepada sebuah <i>Class</i> .

3. Activity Diagram

Activity diagram Merupakan tipe khusus dari *diagram state* yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. *Activity Diagram* secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity Diagram* dapat juga digunakan untuk memodelkan *action* yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari *action* tersebut. *Activity Diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa^[6].

Tabel 2.3 Simbol *Activity Diagram*

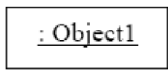



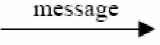
No	Simbol	Nama	Keterangan
1.		<i>Initial Activity</i>	sebagai awal dari aktivitas modul sistem aplikasi.

2.		Activity	menunjukkan aktivitas yang dilakukan.
3.		Final Activity	menunjukkan akhir dari aktivitas.
4.		Decisions	menunjukkan aktivitas yang harus dipilih apakah pilihan pertama atau kedua.
5		Fork	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
6		Rake	Menunjukkan adanya dekomposisi
7			Tanda waktu
8		Flow Final	Aliran akhir
9		Signal	sebagai pengirim dan penerima pesan dari aktivitas yang terjadi. Sinyal terdiri dari sinyal penerima yang digambarkan dengan poligon terbuka dan sinyal pengirim dengan yang digambarkan dengan <i>convex</i> poligon.

4. Sequence Diagram

Sequence Diagram secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah *use case* atau operasi. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima diantara objek dan dalam sekuensi^[6].

Tabel 2.4 Simbol *Sequence Diagram*

No.	Simbol	Nama	Keterangan
1.		Object	<i>Object</i> merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara horizontal.
2.		Actor	<i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom.
3.		Lifeline	<i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam basis waktu.
4.		Activation	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> .
5.		Message	<i>Message</i> digambarkan dengan anak panah horizontal antara <i>activation</i> .

2.2.7 Lampu Lalu Lintas

Lampu lalu lintas (menurut UU no. 22/29 tentang Lalu Lintas dan Angkutan Jalan dan Angkutan Jalan : alat pemberi isyarat lalu lintas atau APILL) adalah lampu yang mengendalikan arus lalu lintas yang terpasang di persimpangan jalan, tempat penyebrangan pejalan kaki (*zebra cross*), dan tempat arus lalu lintas lainnya. Lampu ini yang menandakan kapan kendaraan harus berjalan dan berhenti secara bergantian dari berbagi arah. Pengaturan lalu lintas dipersimpangan jalan dimaksudkan untuk mengatur pergerakan kendaraan pada masing – masing kelompok pergerakan kendaraan agar dapat bergerak secara bergantian sehingga tidak saling mengganggu antar arus yang ada.

1. Sejarah Lampu Lalu Lintas

Lampu lalu lintas terpasang pertama kali di London. Karena pada waktu itu sudah banyak kendaraan motor yang melintas, namun hanya ada 2 warna yaitu hijau dan merah. Lampu ini pertama kali ditemukan oleh Lester Farnsworth Wire. Pada bulan Januari 1896 lampu lalu lintas itu meledak dan melukai salah satu polisi yang sedang berada di dekat lampu lalu lintas itu. Setelah kejadian saat itu, lampu lalu lintas pun tidak dipasang lagi. Namun, setelah itu Garret Augustus Morgan kembali di rancang untuk memperbaiki fungsi dari lampu lalu lintas ini supaya lebih aman, efektif, dan efisien. Pada eksperimennya, Garrett menciptakan lampu lalu lintas yang menambahkan tanda stop dan go untuk mempermudah. Setelah Garret membuat lampu jalan raya yang berwujud seperti huruf T, lampu ini terbagi dalam 3 warna yang berbeda. Warna merah untuk stop, warna hijau untuk go, dan warna kuning untuk posisi stop lainnya. Perbeaan warna kuning dan merah pada stop ini adalah kuning memberi jeda kapan kendaraan berjalan dan mulai berhenti. Lampu kuning juga memberikan kesempatan untuk berhenti dan berjalan secara perlahan. Setelah Lester dan Garret, perkembangan lampu lalu lintas disempurnakan oleh polisi bernama William Potts. Lampu lalu lintas disempurnakan menggunakan tiga warna dan dioperasikan dengan cara otomatis menggunakan tenaga listrik pada Maret 1922.

2. Jenis Lampu Lalu Lintas

a. Berdasarkan Cakupannya :

1) Lampu Lalu Lintas Terpisah

Berdasarkan pada suatu tempat persimpangan saja tanpa mempertimbangkan persimpangan lain.

2) Lampu Lalu Lintas Terkoordinasi

Berdasarkan pertimbangan beberapa persimpangan yang terdapat pada arah tertentu.

3) Lampu Lalu Lintas Jaringan

Berdasarkan pertimbangan beberapa persimpangan yang terdapat dalam suatu jaringan yang masih dalam 1 kawasan.

b. Berdasarkan cara pengoperasiannya :

1) *Fixed Time Traffic Signal*

Lampu lalu lintas yang waktu perubahannya pada waktu yang tepat dan tidak mengalami perubahannya.

2) *Actuated Traffic Signal*

Lampu lalu lintas yang perubahannya pada waktu tertentu dan mengalami perubahan dari waktu ke waktu sesuai dengan waktu kedatangan kendaraan dari berbagai persimpangan. Sistem perubahan waktu tersebut dinamakan logika fuzzy.

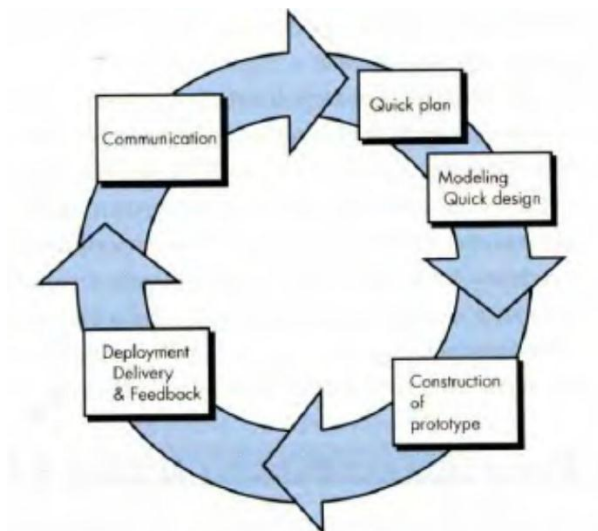
2.2.8 Model Pengembangan Perangkat Lunak

Pada rekayasa perangkat lunak, banyak model yang telah dikembangkan untuk membantu proses pengembangan perangkat lunak. Model-model ini pada umumnya mengacu pada model proses pengembangan sistem yang disebut *System Development Life Cycle* (SDLC). Menurut (Simarmata, 2010) Istilah siklus pengembangan perangkat lunak *Software Development Life Cycle* (SDLC) mengacu pada model dan proses yang digunakan untuk mengembangkan perangkat lunak dan menguraikan proses, yaitu pengembang menerima perpindahan dari *Prototyping* adalah salah satu pendekatan dalam rekayasa perangkat lunak yang secara langsung mendemonstrasikan bagaimana sebuah perangkat lunak atau komponen-

komponen perangkat lunak akan bekerja dalam lingkungannya sebelum tahapan konstruksi aktual dilakukan.

Model *prototype* (*prototyping* model) dapat digunakan untuk menyambungkan ketidakpahaman *client* mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan *client* kepada pengembang perangkat lunak. Sering *client* membayangkan kebutuhan yang diinginkan tapi tidak terspesifikasikan secara detail dari segi masukan, proses, dan keluaran.

Di sisi lain seorang pengembang aplikasi harus menspesifikasi sebuah kebutuhan secara detail dari segi teknis. Metode *prototype* dimulai dari mengumpulkan kebutuhan *client* terhadap aplikasi yang akan dibuat. Kemudian dibuatlah program *prototype* agar *client* lebih terbayang dengan apa yang diinginkan. Program ini biasanya menyediakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti aplikasi yang sudah jadi. Lalu program *prototype* ini dievaluasi oleh *user* sampai ditemukan spesifikasi yang diinginkan *user*^[7].

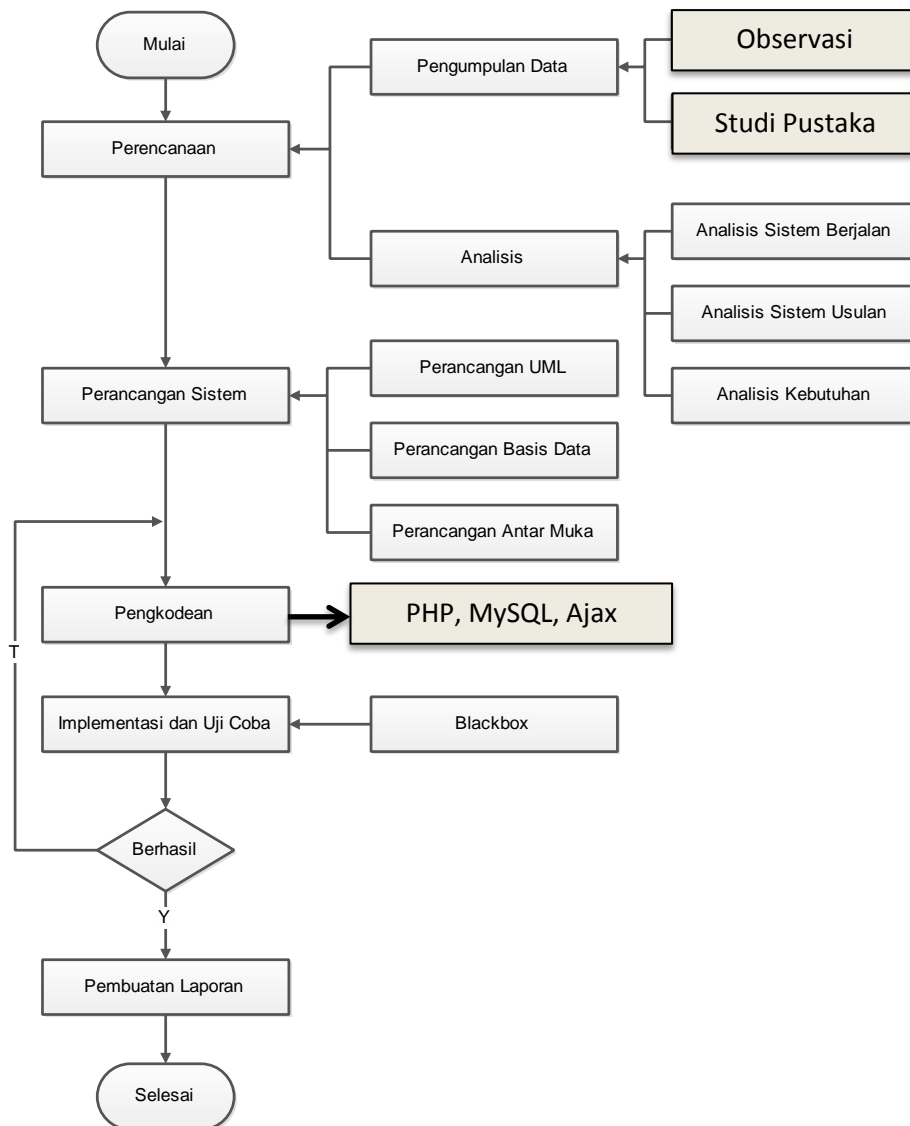


Gambar 2.1 Metode Model **Prototype**

Dalam pembuatan Penyebarluasan Informasi Kepadatan Lalu Lintas Berbasis Web dengan menggunakan metode pengembangan *prototype* memiliki beberapa tahapan yang akan dilakukan yaitu sebagai berikut :

1. Analisis Kebutuhan
2. Pembuatan *Prototype*
3. Evaluasi *prototype*
4. Coding sistem
5. Pengujian sistem
6. Evaluasi sistem
7. Penggunaan sistem permasalahan ke solusi.

2.3 Kerangka pemikiran



Gambar 2.2 Kerangka Pemikiran

Adapun penjelasan dari gambar kerangka pemikiran diatas adalah sebagai berikut :

1. Perencanaan

Pada tahap ini penulis melakukan perencanaan terhadap topik apa yang akan diangkat pada penelitian yang akan dilakukan untuk penyusunan skripsi ini. Tahap perencanaan dimulai dengan melakukan pengumpulan data untuk mendapatkan informasi yang dibutuhkan dalam penelitian ini. Teknik pengumpulan data yang digunakan dengan cara observasi pada jalan Matraman 1 Jakarta Timur dan dengan studi pustaka. Selanjutnya dilakukan juga analisis terhadap sistem yang sekarang berjalan di Lampu Lalu Lintas Matraman 1 Jakarta Timur. Dari analisis tersebut akan didapat kelemahan sistem dan masalah yang timbul dari sistem yang berjalan tersebut. Sehingga penulis dapat menganalisa sistem yang dibutuhkan untuk diusulkan membantu pemecahan masalah dari sistem yang berjalan. Dan juga penulis melakukan analisis kebutuhan dari website dan aplikasi yang dibangun yang meliputi kebutuhan pengguna, kebutuhan input, kebutuhan output, dan kebutuhan perangkat penelitian.

2. Perancangan Sistem

Tahap selanjutnya setelah analisis adalah tahap perancangan sistem. Pada tahap perancangan sistem, penulis mendefinisikan apa yang diperlukan dari kebutuhan sistem, melakukan persiapan untuk merancang bangun implementasi dan menggambarkan bagaimana suatu sistem dibentuk. Perancangan yang dilakukan oleh penulis terdiri dari perancangan UML, perancangan basis data, dan perancangan antar muka yang terkait pada aplikasi.

3. Pengkodean

Pengkodean merupakan tahap yang dilakukan setelah perancangan sistem terbentuk, di mana mengimplementasikan semua masalah yang ada ke dalam program yang nantinya dapat digunakan untuk pemecahan masalah yang dihadapi. Selanjutnya pengkodean untuk pembuatan aplikasi, yang mana hasil perancangan diterjemahkan menjadi bentuk yang dapat dibaca

atau dimengerti oleh komputer, berupa bahasa pemrograman php dan mysql.

4. Implementasi dan Uji Coba

Tahapan selanjutnya yaitu melakukan implementasi dan uji coba aplikasi. Pengujian web ini menggunakan pengujian black box. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Apabila web tidak dapat berjalan dengan baik, maka akan dicari kesalahan yang nantinya akan diperbaiki sampai web dapat berjalan dengan baik. Dilakukan uji coba juga untuk mengetahui jika aplikasi mengalami kegagalan atau error maka akan dicek kembali dengan menganalisa kode program yang salah, kemudian evaluasi program.

5. Pembuatan Laporan

Setelah aplikasi telah berhasil berjalan dengan baik seperti yang diharapkan, maka akan dilakukan dokumentasi. Dokumentasi dilakukan terhadap jalannya aplikasi pada penyebaran informasi kepadatan lalu lintas secara realtime. Kemudian penulis akan membuat laporan terhadap aplikasi yang telah dibangun dan dijalankan.